

Anwendergesteuerte Aufbereitung von Bibliographien auf der Basis von XML, XSL und PHP

Diplomarbeit

im Fach Informationsnetze, Kommunikationstechnik und
Netzwerkmanagement
Studiengang Informationsmanagement
der
Fachhochschule Stuttgart –
Hochschule der Medien

Nora Kersten

Erstprüfer:	Prof. Dr. Wolf-Fritz Riekert
Zweitprüfer:	Peter Thies

Bearbeitungszeitraum: 15. Juli bis 15. Oktober 2002

Stuttgart, Oktober 2002

Kurzfassung

In dieser Arbeit wird demonstriert, wie Bibliographien, die im XML-Format vorliegen, nach unterschiedlichen Anwenderanforderungen für die Präsentation im Web aufbereitet werden können. Zu diesem Zweck wird ein kurzer Überblick der Auszeichnungssprache XML gegeben, im Besonderen des für Bibliographien entwickelten Formats BibTeXML. Es wird ein Softwareprototyp entworfen und entwickelt, der die anwendergesteuerte Aufbereitung von Bibliographien ermöglicht, die im BibTeXML-Format vorliegen. In XSL kodierte Aufbereitungsfunktionen ermöglichen verschiedene Optionen der Anzeige, der Sortierung und der Auswahl der darzustellenden Literaturangaben. Der Softwareprototyp, ein PHP-Skript, erlaubt die Einstellung der verfügbaren Optionen und den Start des Aufbereitungsprozesses. Die entwickelten Skripte lassen sich in eine Website (z. B. Homepages von Professoren oder Fachbereichen) integrieren und die Ergebnisse in jedem Browser anzeigen.

Schlagwörter: Bibliographie, Literaturverzeichnis, BibTeXML, PHP, Anzeige

Abstract

This thesis shows how to edit bibliographies existing as BibTeXML files for a presentation in the WWW according to different user requirements. A short overview gives information on the markup language XML, especially on the XML file format for bibliographies, BibTeXML. A software prototype is designed and worked out that supports user-controlled editing of bibliographies existing in BibTeXML. Editing functions that are programmed in XML make different options possible for displaying, sorting and selecting the reference list. A PHP-Script allows the setting of available options and starts the editing process. These scripts are designed to be integrated in a home page (for example in an university teacher's or an university department's home page) and the results can be viewed in every web-browser.

Keywords: bibliography, reference list, BibTeXML, PHP, display

Inhaltsverzeichnis

Kurzfassung.....	2
Abstract.....	2
Inhaltsverzeichnis	3
Abbildungsverzeichnis	5
Tabellenverzeichnis.....	5
Abkürzungsverzeichnis	6
Vorwort.....	7
1 Überblick.....	8
2 Aufgabenstellung und Zielsetzung.....	10
3 Technologien und Beispiele für die webbasierte Anzeige von Bibliographien	11
3.1 Von XML, Tex und Latex zu BibTeXML	11
3.1.1 XML.....	11
3.1.2 Tex und Latex.....	12
3.1.3 BibTeXML.....	12
3.2 XSL	13
3.3 PHP	13
3.4 Bibliographiesammlung von „The Collection of Computer Science Bibliographies“	13
3.5 Bibliographie des Real-Time Systems Laboratory.....	16
3.6 Diplomarbeit von Heike Müller	17
3.7 The Bibtex-XML-HTML Project	18
4 Zu unterstützende Darstellungsformate für Bibliographien	20
4.1 Übersicht	20
4.2 Anzeigestile	20
4.2.1 Anzeigestil nach DIN 1505 Teil 2	21
4.2.2 Anzeigestil nach APA (American Psychological Association).....	24
4.2.3 Anzeigestil nach Harvard	25
5 Konzeption der Bibliographieanzeige	28
5.1 Übersicht	28
5.2 Anzeige im Internet.....	28
5.3 Suchfunktion.....	28

5.4	Sortierfunktion	29
5.5	Auswahl des Anzeigestils	29
6	Benutzung aus Sicht des Anwenders	30
6.1	Übersicht	30
6.2	Suchformular	30
6.3	Suchbeispiele	31
6.3.1	Suche nach Autornamen	31
6.3.2	Verknüpfung von Suchbegriffen.....	33
6.4	Anzeige der gesamten Bibliographie	35
6.5	Sortierung.....	37
6.6	Verwendung in einer Worddatei.....	39
6.7	Einbindung in eine Homepage.....	39
7	Realisierung	40
7.1	Übersicht	40
7.2	Interne Abfolge der Programmteile	40
7.3	PHP-Formular.....	42
7.4	Aufruf des XSLT-Prozessors	45
7.5	Verwendung der XSL-Stylesheets	46
7.5.1	Grundlagen.....	46
7.5.2	Realisierung der Sortierfunktion.....	49
7.5.3	Anzeige eines oder mehrerer Autornamen	50
7.5.4	Anzeige der übrigen Teile einer Quellenangabe	52
7.5.5	Umsetzung der Suchfunktion	54
7.6	Ausgabe der Ergebnisliste	58
8	Zusammenfassung und Ausblick.....	59
Anhang A	60	
Beispielcode einer BibTeXXML-Datei		60
Beispielcode einer DTD für BibTeXXML		62
Anhang B	66	
Beispielcode der XSL-Datei für den Harvard-Anzeigestil		66
Anhang C	99	
Kompletter Code des PHP-Formulars.....		99
Literaturverzeichnis.....	103	
Printmedien		103
Elektronische Ressourcen		103
Erklärung.....	105	

Abbildungsverzeichnis

Abbildung 1: Collection of Computer Science Bibliographies: Suchformular	14
Abbildung 2: Collection of Computer Science Bibliographies: Ergebnisanzeige im Zitier-Format	14
Abbildung 3: Collection of Computer Science Bibliographies: Vergrößerung der Ergebnisliste im Zitierformat	15
Abbildung 4: Collection of Computer Science Bibliographies: Ergebnisanzeige im Bibtex-Format.....	15
Abbildung 5: Real-Time-Systems Laboratory: Suchformular der Bibliographie	16
Abbildung 6: Real-Time-Systems Laboratory: Ergebnisliste.....	16
Abbildung 7: Real-Time-Systems Laboratory: Vergrößerung eines Teils des Suchergebnisses.....	17
Abbildung 8: Real-Time-Systems Laboratory: Suchergebnis mit AND-Verknüpfung ...	17
Abbildung 9: Bibtex-XML-HTML Project: Übersichtsdiagramm der erstellten HTML- Seiten.....	19
Abbildung 10: Suchformular.....	30
Abbildung 11: Suche nach Autor	32
Abbildung 12: Suche nach Autor - Suchergebnis (oberer Teil).....	32
Abbildung 13: Suche nach Autor - Suchergebnis (unterer Teil).....	33
Abbildung 14: Suchergebnis für Autor und Jahr.....	33
Abbildung 15: Suchergebnis für Autor, Jahr und Literaturart.....	34
Abbildung 16: Suchformular für die Ausgabe der gesamten Bibliographie	35
Abbildung 17: Ausgabe der gesamten Bibliographie – oberer Teil.....	36
Abbildung 18: Ausgabe der gesamten Bibliographie - unterer Teil.....	36
Abbildung 19: Sortierung aufsteigend nach Autor	37
Abbildung 20: Sortierung absteigend nach Autor	37
Abbildung 21: Sortierung aufsteigend nach Jahr.....	38
Abbildung 22: Sortierung absteigend nach Jahr.....	38
Abbildung 23: Auswertung der Formulareingaben	41
Abbildung 24: Ausgabe der Ergebnisliste	41
Abbildung 25: Beispiel einer Baumstruktur von BibTeXXML	48
Abbildung 26: Darstellung der Struktur der Suchfunktion.....	55

Tabellenverzeichnis

Tabelle 1: nach Lorenzen 1997, S. 8. : Monographien, Sammelwerke, Forschungsberichte usw.	22
Tabelle 2: nach Lorenzen 1997, S. 13: Aufsätze in Zeitschriften und Zeitungen	23

Abkürzungsverzeichnis

APA	American Psychological Association
ASCII	American Standard Code for Information Interchange
DIN	Deutsches Institut für Normung, Deutsche Industrienormen
DTD	Document Type Definition, Dokumenttyp-Definition
HTML	Hypertext Markup Language, Auszeichnungssprache
PHP	PHP Hypertext Preprocessor, Skriptsprache
XML	eXtensible Markup Language, Auszeichnungssprache
XSL	eXtensible Stylesheet Language, Transformations- und Formatierungssprache

Vorwort

Diese Arbeit basiert auf den Vorarbeiten, die in der Diplomarbeit von Heike Müller (2001) zum Thema Anzeigemöglichkeiten von BibTeX-XML-Bibliographien gemacht wurden. Zusätzlich konnte ich mich auf eine weitere Diplomarbeit stützen, die Sebastian Spaleck im gleichen Jahr an der Hochschule der Medien, Stuttgart vorlegte. Diese Diplomarbeit war für die Entwicklung des PHP-Skripts für den XSL-Prozessor-Aufruf sehr hilfreich.

Ein Dankeschön für die Unterstützung bei der Entstehung dieser Arbeit möchte ich Herrn David Prüm von der HdM Stuttgart für die Einrichtung und Hilfe mit dem Sablotron-Prozessor und Herrn Erik Wilde für seine hilfreichen Hinweise zu XSL aussprechen.

1 Überblick

In einer Bibliographie am Ende einer wissenschaftlichen Arbeit werden die Angaben aller Quellen aufgelistet, die in der Arbeit zitiert oder verwendet wurden.

Um eine Bibliographie zu erstellen, benötigt man die bibliographischen Angaben der einzelnen Quellen, die man auflisten möchte, sowie eine Strukturierungsanleitung, die ein einheitliches Erscheinungsbild für alle Angaben ermöglicht. Die Strukturierungsanleitung enthält Regeln für die Darstellung, die je nach Art der Quelle unterschiedlich sein kann. Außerdem gibt sie an, in welcher Reihenfolge die Quellenangaben sortiert werden sollen und ermöglicht es, schnell zu erkennen, um welche Art von Quelle es sich handelt.

Es gibt je nach der Zitierweise, die in der wissenschaftlichen Arbeit verwendet wurde, verschiedene Darstellungsformate für die Bibliographie. Dazu gehören unter anderem die Darstellung der Quellenangaben nach MLA¹, APA (American Psychological Association 2001), Chicago², Harvard, DIN 1505 Teil 2 oder Turabian³.

Im wissenschaftlichen Bereich werden Arbeiten häufig mit der Auszeichnungssprache Tex erstellt. Ein spezielles Makroprogramm für Bibliographien im Tex-Format heißt "Bibtex".

Aus Bibtex-Dateien lassen sich verschiedene Arten von BibTeX-Dateien erstellen, einem XML-Format, das die Vorteile der Tex-Bibliographien, wie z. B. ihre hohe Verfügbarkeit, mit der Strukturierung und den Anpassungsmöglichkeiten von XML verbindet.

Die Verwendung von XML-Dateien ist heute vorwiegend im Internet angesiedelt, wo sie zur Anzeige in einem Browser, wie z. B. Internet Explorer oder Netscape Navigator, als HTML-Ausgabe aufbereitet werden. Neben dem Einsatz im Internet gibt es aber auch andere Verwendungsformen. So ist es zusätzlich möglich, eine XML-Datei derart zu verarbeiten, dass sie in einem anderen Format als dem HTML-Format ausgegeben wird. Andere Ausgabeformate sind zum Beispiel das einer Textdatei oder einer weiteren, unterschiedlich aufgebauten XML-Datei. Dadurch ermöglicht die Benutzung von XML-Dateien zusätzlich einen einfach zu realisierenden Datenaustausch zwischen verschiedenen Servern und Anwendungen, der sich nicht nur auf das Internet beschränkt.

¹ GIBALDI, JOSEPH: *MLA handbook for writers of research papers*. New York : Modern Language Association of America, 1995

² *The Chicago Manual of Style*. 14th ed. Chicago: University of Chicago Press, 1993

³ TURABIAN, KATE L. : *A Manual for Writers of Term Papers, Theses, and Dissertations*. 6th ed., Chicago : University of Chicago Press, 1996

Um eine Bibliographie im Internet anzuzeigen, die vom Anwender verändert werden kann, werden verschiedene Technologien benötigt. In dieser Diplomarbeit wird eine mögliche Lösung für dieses Problem aufgezeigt. Im zweiten Kapitel wird die Aufgabenstellung noch einmal genau dargestellt. Im dritten Kapitel folgt ein Überblick über die verwendeten Technologien und es enthält eine Übersicht über bereits existierende Darstellungslösungen.

In Kapitel 4 wird beschrieben, wie die Darstellungsregeln unterschiedlicher Anzeigestile die Darstellung für verschiedene Arten von Quellen definieren. Es wird genauer auf die Darstellungsvorschriften von DIN 1505 Teil 2, APA und Harvard eingegangen.

Im fünften Kapitel wird ein Überblick über die Anforderungen eines Anzeigetools gegeben.

Das sechste Kapitel enthält Anwendungsbeispiele des entwickelten Anzeigetools. Es wird eine Erklärung für die Benutzung der Suchfunktionen und Darstellungsstile gegeben. Zusätzlich wird erklärt, auf welche Weise das Tool vom Benutzer verwendet werden kann.

Die zugehörigen Codebeispiele und technischen Grundlagen für die Bibliographieanzeige liefert das siebte Kapitel. Die Anzeige von BibTeXML-Bibliographien in einem Internetbrowser ist erst nach der Verarbeitung mit XSL-Stylesheets in einem XSL-Prozessor möglich. Um die Anzeige einer BibTeXML-Datei in einem Browser nach Anwendereingaben ändern zu können, benötigt man zusätzlich ein PHP-Skript, das Eingaben verarbeiten und Teile der XSL-Datei für die gewünschte Ausgabe anpassen kann. In Kapitel 7 wird neben der Programmierung des PHP-Formulars auch der Aufbau und die Sortier- und Suchfunktionen der XSL-Stylesheets erklärt.

2 Aufgabenstellung und Zielsetzung

Um eine Bibliographie, die im BibTeX-Format vorliegt, in einem Ergebnisformat darzustellen, das die meisten Anwender problemlos anzeigen und benutzen können, bietet sich die Verarbeitung zum HTML-Format an. Bei diesem Format lassen sich leicht Elemente einbinden, die ein Anwender mit seinen Eingaben beeinflussen kann. So kann er mit einem Formular durch das Ausfüllen von Feldern Kriterien festlegen, nach denen eine XML-Datei mit den entsprechenden Veränderungen angezeigt wird.

Ziel dieser Arbeit ist es, dem Anwender mit Hilfe einer Webapplikation eine Auswahl an Optionen bereitzustellen, mit der er die Ausgabe des HTML-Formats aus der Bibliographiedatei, die im BibTeX-Format vorliegt, beeinflussen kann.

Um diese anwendergesteuerte Aufbereitung zu realisieren wird ein Formular erstellt, das in einem Internetbrowser angezeigt werden kann. Eine benutzerfreundliche Suchfunktion, die in der Anzeige integriert ist, ermöglicht es dem Anwender, nur gewünschte Ausgaben zu erzeugen.

Das Formular enthält auch eine Funktion zur Einschränkung auf bestimmte Arten von Quellen, wie z. B. Zeitschriftenaufsätze oder Bücher. Hinzu kommt eine Auswahlfunktion für den Stil der angezeigten Quellenangaben (z. B. nach den Regeln der Norm DIN 1505 Teil 2). Die Suchfunktion ermöglicht es, verschiedene Einschränkungen für die angezeigten Quellenangaben vorzunehmen, die z. B. Autornamen oder Titel der Quelle betreffen.

Nach der Eingabe der Auswahlkriterien durch den Benutzer wird eine Ergebnisliste im HTML-Format erzeugt, die im Browser angezeigt wird. Diese HTML-Ausgabe ändert sich flexibel je nach den Formulareingaben des Anwenders.

Es soll ermöglicht werden, diese Anwendung mit ihren Funktionalitäten in eine andere Anwendung einzubinden, zum Beispiel in eine Internetseite. Dafür würden sich Homepages wie die eines Professors oder eines Fachbereichs besonders anbieten.

3 Technologien und Beispiele für die webbasierte Anzeige von Bibliographien

3.1 Von XML, Tex und Latex zu BibTeXML

3.1.1 XML

Daten, die im XML-Format vorliegen, müssen meist erst aufbereitet werden, bevor sie sinnvoll genutzt werden können. Allerdings bietet das XML-Format einige Vorteile, die diesen Nachteil schnell überwiegen: XML-Dateien sind plattformunabhängig. Erst durch die Weiterverarbeitung durch einen Prozessor mit speziellen Sprachen und Programmen wie XSLT oder XPath werden Anpassungen vorgenommen, die die Darstellung auf verschiedenen Systemen erlauben. Beispiele für den plattformübergreifenden Einsatz von XML sind nach Goldfarb (Goldfarb 2002) neben dem Austausch von Informationen und Inhalten (Content) von verschiedenen Quellen bei webbasierten Abodiensten (Goldfarb 2002, S. 359 ff) auch im Bereich der Unternehmenskommunikation anzusiedeln. XML ermöglicht es beispielsweise, das Spezialvokabular eines Unternehmens einem anderen Unternehmen, wie z. B. einem Kunden oder einem Zulieferer, zugänglich zu machen (Goldfarb 2002, S. 571 ff).

Zudem ist XML je nach Bedarf erweiterbar und leicht auf spezielle Anforderungen anzupassen. Es besitzt ein einfaches, robustes Datenformat (ASCII), das wahrscheinlich auch in Zukunft eine Konversion in ein anderes Format unnötig macht und ist im Grunde mit seiner Baumstruktur selbstbeschreibend. Komplexe Dokumente werden strukturiert, was den Zugriff selbst bei großen Dateien erleichtert. Daten können auch aus mehreren Quellen stammen, die das XML-Dokument zusammenfügt (Harold 2002, S. 33-36).

Zu einer wohlgeformten und validen XML-Datei gehört eine DTD (Document Type Definition, Dokumenttyp-Definition). Sie legt fest, welche Elemente, Entitäten oder Attribute in der XML-Datei vorkommen dürfen. Außerdem lässt sich in der DTD festlegen, wie die Baumstruktur aufgebaut sein soll, welche Arten von Zeichen für Element- oder Attributinhalt erlaubt sind (festgelegt durch z. B. CDATA, PCDATA, ID, IDREF), und welche Eigenschaften die einzelnen Teile haben sollen, z. B. Vorgaben für Attributwerte oder für Elemente, die angegeben werden müssen (#REQUIRED), nicht unbedingt angegeben werden müssen (#IMPLIED) oder eindeutig festgelegt sind (#FIXED).

3.1.2 Tex und Latex

Im wissenschaftlichen Bereich werden Arbeiten häufig mit der Auszeichnungssprache Tex erstellt. Sie unterscheidet sich beispielsweise von Microsofts Office-Programm "Word" darin, dass für Textelemente nur festgelegt wird, zu welcher Art sie zugehörig sind (z. B. Titel oder Textabsatz). In einem Makroprogramm - für den wissenschaftlichen Bereich wird meist "Latex" verwendet - sind alle Anweisungen zum Layout gespeichert. So genügt es, in der Tex-Datei den reinen Inhalt mit den Elementdefinitionen aufzuschreiben. Die einheitliche Formatierung erfolgt erst durch die Anwendung des Makros. Danach wird der Text in der endgültigen Fassung dargestellt. Tex besitzt auch ein spezielles Makroprogramm für Bibliographien, das "Bibtex" heißt und von Oren Patashnik entwickelt wurde (Patashnik 1988).

Hier folgt ein Beispiel für eine Quellenangabe als Teil einer Bibtex-Datei (aus einer Bibtex-Datei des Real-Time Systems Laboratory (2001, <http://pertsserver.cs.uiuc.edu/papers/rtsl.bib> , Zugriff am 3.10.2002)):

```
@InCollection{Bettati91a,
  author =    {R. Bettati and D. Gillies and C. C. Han and
               K. J. Lin and C. L. Liu and J. W. S. Liu and W. K. Shih},
  title =    {Recent Results in Real-Time Scheduling},
  crossref =    {vanTilborg91a},
  chapter =    4,
  pages =    {91-127},
  citesrc =    {dlh}
}
```

3.1.3 BibTeXML

Bibtex-Bibliographien besitzen durch die eindeutige Festlegung der einzelnen Elemente, die für die Tex-Funktionalität benötigt wird, eine stark strukturierte Form, die leicht umgewandelt werden kann. Dadurch bieten sie eine gute Grundlage für eine Verarbeitung mit XML. Das Format einer XML-Datei, die aus einer Bibtex-Bibliographie umgewandelt wurde, heißt BibTeXML. Wie die Diplomarbeiten von Brenno Lurati und Luca Previtali (Lurati 2001) oder von Heike Müller (Müller 2001) und die Website von Marco Kuhlmann (Kuhlmann 2002) zeigen, lassen sich aus Bibtex-Dateien verschiedene Arten von BibTeXML-Dateien erstellen. BibTeXML unterstützt die Baumstruktur von XML und läßt sich mit allen für XML entwickelten Funktionen wie z. B. XSLT, XPath, XPointer oder XLink weiterverarbeiten. Ein Beispiel für eine BibTeXML-Datei und ihre DTD ist im Anhang A aufgeführt. In Kapitel 7 wird der Aufbau der Struktur näher erläutert.

3.2 XSL

Die Transformations- und Formatierungssprache für XML-Dateien dient dazu, sie für eine Darstellung zu formatieren. Bei der Verarbeitung generiert der Prozessor aus den Transformationsanweisungen der XSL-Datei und den verwendeten XML-Daten eine Baumstruktur, den Ergebnisbaum. Zusätzlich wendet der Prozessor die XSL-Anweisungen zur Formatierung an den Ästen und Knoten des Ergebnisbaumes an.

Das XSL-Stylesheet enthält auch die Anweisung, was für ein Format ausgegeben werden soll. Es ist neben HTML auch möglich, den Ergebnisbaum als Text oder XML auszugeben.

In XSL-Templates wird für alle XML-Elemente festgelegt, welche Daten der XML-Datei an welcher Stelle und mit welcher Formatierung dargestellt werden. Zu XSL gehören die Spezifikationen für XSL-FO (XSL Formatting Objects), XSLT (XSL Transformations), und XPath. XSL-FO (XSL Formatting Objects) ermöglicht ausgefeilte Formatierungsformen. XSLT (XSL Transformations) beinhaltet Anweisungen für die Verarbeitung im Prozessor. XPath ermöglicht die gezielte Ansteuerung von Knoten in der Baumstruktur

Der Aufbau einer XSL-Datei ist im Anhang B zu finden.

3.3 PHP

PHP (PHP Hypertext Preprocessor) ist eine Skriptsprache, die serverseitig Programme verarbeiten kann. Der Code wird im Unterschied zu den meisten CGI-Programmiersprachen in den HTML-Code selbst eingebettet. Seit ihrer Erfindung im Jahr 1995 wurde PHP immer weiterentwickelt, bis sie heute neben anderen CGI-Sprachen, wie z. B. Perl, zu einem der wichtigsten Mittel zur dynamischen Seitengenerierung im Web wurde. PHP bietet eine große Anzahl von Programmfunktionen und Routinen, zu denen laufend neue hinzukommen.

3.4 Bibliographiesammlung von „The Collection of Computer Science Bibliographies“

<http://iinwww.ira.uka.de/bibliography> (Zugriff am 3.10.2002)

Auf dem Server des Lehrstuhls Informatik für Ingenieure und Naturwissenschaftler in Karlsruhe hat Alf-Christian Achilles eine große Sammlung von Bibliographien im Bibtex-Format (ca. 1400) veröffentlicht. Möglich ist eine Suche nach einer Bibliographiedatei sowohl nach Kategorie, als auch nach verschiedenen Suchoptionen.

Bei einem Ergebnis werden Metadaten zur gefundenen Bibliographie angezeigt. Die Metadaten enthalten neben einer Beschreibung der vorhandenen Felder auch einen Link zur Originalbibliographie im Bibtex-Format. Die gefundene Bibtex-Datei selbst ist nach verschiedenen Optionen durchsuchbar. Ausgewählt werden können Einschränkungen

nach Groß- und Kleinschreibung, Suche nach Zeichenketten aus einem oder mehreren Wörtern.

Software Product Line Bibliography

[[About](#) | [Browse](#) | [Statistics](#)]

Number of references: 660

Number of online publications: 60

Most recent reference: August 2002

Last update: September 27, 2002

Supported: yes

Search the Bibliography

Query:

Options: case insensitive partial word(s) ☐ online papers only

Results: Citation Maximum of matches

Abbildung 1: Collection of Computer Science Bibliographies: Suchformular

Man kann die Ergebnisanzeige wahlweise im Zitierformat oder im originalen Bibtex-Format ausgeben lassen.

Simple Search Results:

You have searched: "[Software Product Line Bibliography](#)".

Query: design

Options: case-insensitive; partial words match; searching matches 1 to 40.

From [Software Product Line Bibliography](#):

- [O.-A. Agyapong](#) (HP) and [P. Bobbie](#) (HP)

The Design of an Expert System for Domain Knowledge Engineering and Decision Making: A Case Study in the Criminal Justice System

International Journal of Software Engineering & Knowledge Engineering, 8(1), pp. 21-33, March 1998.

[BibTeX](#)
- [Pierre America](#) (HP) and [Jan van Wijgerden](#) (HP)

Requirements Modeling for Families of Complex Systems

Software Architectures for Product Families. Proceedings of the Third International Workshop on Software Architectures for Product Families, LNCS 1951, pp. 199-209, Springer, March 2000.

[BibTeX](#)
- [G. Arango](#) (HP)

Notes on the Application of the COBWEB Clustering Function to the Identification of Patterns of Reuse

Fifth International Workshop on Software Specification and Design, 1988.

[BibTeX](#)
- [G. Arango](#) (HP)

Domain Analysis -- From Art Form To Engineering Discipline

Fifth International Workshop on Software Specification and Design, pp. 152-159, September 1989.

[BibTeX](#)
- [G. Arango](#) (HP) and [E. Schoen](#) (HP) and [R. Pettengill](#) (HP)

A Process for Consolidating and Reusing Design Knowledge

Proceedings of the Fifteenth International Conference on Software Engineering (ICSE'93), pp. 231-242, 1993.

[BibTeX](#)
- [M. Ardis](#) (HP) and [D. Weiss](#) (HP)

[BibTeX](#)

Abbildung 2: Collection of Computer Science Bibliographies: Ergebnisanzeige im Zitier-Format

From [Software Product Line Bibliography](#).

- [O.-A. Agyapong](#) (HP) and [P. Bobbie](#) (HP)
The Design of an Expert System for Domain Knowledge Engineering and Decision Making: A Case Study Justice System
International Journal of Software Engineering & Knowledge Engineering, 8(1), pp. 21-33, March 1998.
- [Pierre America](#) (HP) and [Jan van Wijgerden](#) (HP)
Requirements Modeling for Families of Complex Systems
Software Architectures for Product Families. Proceedings of the Third International Workshop on Software Architectures for Product Families, LNCS 1951, pp. 199-209, Springer, March 2000.
- [G. Arango](#) (HP)
Notes on the Application of the COBWEB Clustering Function to the Identification of Patterns of Reuse
Fifth International Workshop on Software Specification and Design, 1988.
- [G. Arango](#) (HP)
Domain Analysis -- From Art Form To Engineering Discipline
Fifth International Workshop on Software Specification and Design, pp. 152-159, September 1989.
- [G. Arango](#) (HP) and [E. Schoen](#) (HP) and [R. Pettengill](#) (HP)
A Process for Consolidating and Reusing Design Knowledge
Proceedings of the Fifteenth International Conference on Software Engineering (ICSE'93), pp. 231-242, 1993

Abbildung 3: Collection of Computer Science Bibliographies: Vergrößerung der Ergebnisliste im Zitierformat

From [Software Product Line Bibliography](#).

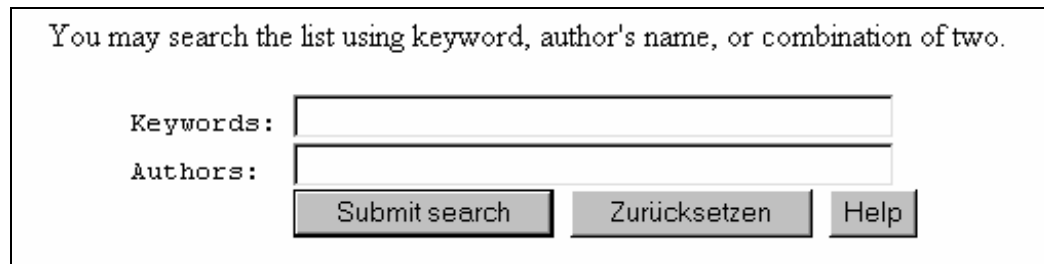
```
@Article{Agyapong:1998:DKE,
  author =      "O.-A. Agyapong and P. Bobbie",
  title =       "(The Design of an Expert System for Domain Knowledge
                  Engineering and Decision Making: A Case Study in the
                  Criminal Justice System)",
  abstract =    "In this paper, we describe a tool for eliciting domain
                  knowledge (specification) of a decision support system.
                  In particular, we focus on a decision support system
                  (DSS) which employs domain knowledge of recidivism in
                  the juvenile justice system. Using the elicited domain
                  knowledge, the DSS tool uses deductive reasoning
                  techniques to make inferences and provide suggestive
                  courses of action to support the investigatory
                  functions of police, attorneys, or probation officers.
                  (...)",
  journal =     "International Journal of Software Engineering \&
                  Knowledge Engineering",
  year =        "1998",
  month =       mar,
  volume =      "8",
  number =      "1",
```

Abbildung 4: Collection of Computer Science Bibliographies: Ergebnisanzeige im Bib-tex-Format

3.5 Bibliographie des Real-Time Systems Laboratory

<http://pertsserver.cs.uiuc.edu/papers/index.html>

Die Bibliographie basiert auf einer Bibtex-Datei, die die Publikationen der Mitglieder des Real-Time Systems Laboratory enthält. Die Bibliographie ist in einem Browser anzeigbar und lässt sich durchsuchen.



You may search the list using keyword, author's name, or combination of two.

Keywords:

Authors:

Abbildung 5: Real-Time-Systems Laboratory: Suchformular der Bibliographie

Auf der Startseite befindet sich unter dem Suchformular eine Auflistung aller Bibliographieinträge. Es ist möglich, Autornamen und ein Stichwort in das Suchformular einzugeben. Die Anzeige der Ergebnisliste erfolgt unsortiert.

Search results for keyword 'design'

Search performed on <http://pertsserver.cs.uiuc.edu/papers/index.html>

- R., Lui Sha Gagliardi, M.; Rajkumar. Designing for evolvability: building blocks for evolvable real-time systems. In *Proceedings of the 2nd IEEE Real-Time Technology and Applications Symposium (RTAS '96)*, June 1996.
- T. Kim, C. L. Liu, and J. W. S. Liu. A scheduling algorithm for conditional resource sharing--a hierarchical reduction approach. *IEEE Transactions on Computer-Aided Design*, 13(4):425-438, April 1994.
- R. Rajkumar, M. Gagliardi, and L. Sha. The real-time publisher /subscriber inter-process communication model for distributed real-time systems: Design and implementation. In *First IEEE RealTime Technology and Applications Symposium*, May 1995.
- K. rz, A. Cervin, J. Eker, and L. Sha. An introduction to control and real-time scheduling co-design. In *Proceedings of the 39th Conference on Decision & Control*, December 2000.
- L. Sha and S. Sathaye. Distributed real-time system design using generalized rate monotonic theory. In *Proceedings of the Second International Conference on Automation, Robotics, and Computer Vision*, September 1992.
- Lui Sha and Shirish S. Sathaye. A systematic approach to designing distributed real-time systems. *IEEE Computer*, 26(9):68-79, 1993.
- Ami Abraham Silberman. *RTM--Design and Implementation*. PhD thesis, University of Illinois at Urbana-Champaign, 1997. Available as Technical Report UIUCDCS-R-97-2040. ([Gzipped PostScript](#), 268 pages, 568938 bytes) ([PDF](#), 798364 bytes)
- X. Song and J. W. S. Liu. Maintaining temporal consistency: Pessimistic vs. optimistic concurrency control. In *Proceedings of the IEEE Symposium on Computer-Aided Control System Design*, Napa, California, March 1992. ([PostScript](#), 28 pages, 508132 bytes)
- Matthew F. Storch and Jane W.-S. Liu. DRTSS: A simulation framework for complex real-time systems. In *Proceedings of the Complex Systems Design*

Abbildung 6: Real-Time-Systems Laboratory: Ergebnisliste

- R.; Lui Sha Gagliardi, M.; Rajkumar. Designing for evolvability: building blocks for evolvable real-time *Technology and Applications Symposium (RTAS '96)*, June 1996.
- T. Kim, C. L. Liu, and J. W. S. Liu. A scheduling algorithm for conditional resource sharing--a hierarchical reduction approach. *Computer-Aided Design*, 13(4):425-438, April 1994.
- R. Rajkumar, M. Gagliardi, and L. Sha. The real-time publisher /subscriber inter-process communication implementation. In *First IEEE RealTime Technology and Applications Symposium*, May 1995.
- K. rzn, A. Cervin, J. Eker, and L. Sha. An introduction to control and real-time scheduling co-design. *& Control*, December 2000.

Abbildung 7: Real-Time-Systems Laboratory: Vergrößerung eines Teils des Suchergebnisses

Die Anzeige erfolgt soweit feststellbar nicht nach den Stilvorschriften von DIN 1505, APA oder Harvard.

Eine gleichzeitige Eingabe von Autor und Keyword schränkt die oben gezeigte Liste mit einer AND-Verknüpfung ein. In der Beispielsuche erscheint nur ein einziger Ergebniseintrag:

Search results for author `kim' keyword `design'

Search performed on <http://pertserver.cs.uiuc.edu/papers/index.html>

- T. Kim, C. L. Liu, and J. W. S. Liu. A scheduling algorithm for conditional resource sharing--a hierarchical reduction approach. *IEEE Transactions on Computer-Aided Design*, 13(4):425-438, April 1994.

Abbildung 8: Real-Time-Systems Laboratory: Suchergebnis mit AND-Verknüpfung

3.6 Diplomarbeit von Heike Müller

Heike Müller hat in ihrer Diplomarbeit Möglichkeiten beschrieben, wie BibTeXML-Dateien gemäß der Darstellungsvorschrift von DIN 1505 Teil 2 angezeigt werden können. Neben der Definition von einzelnen Anzeigeoptionen für Quellenarten wie Zeitschriftenaufsätze oder Bücher durch XSL-Stylesheets (Müller 2001, S. 29 ff) wird auch ein Beispiel zur Formatierung von Autornamen für die Anzeige in einem Webbrowser gegeben. Zudem wird die alphabetische Sortierung der gesamten Bibliographie in der Anzeige durch XSL-Stylesheets erklärt (Müller 2001, S. 35). Es wird vorgeschlagen, die nach der Verarbeitung entstandene HTML-Ausgabe als Datei in ein Word-Dokument einzufügen (Müller 2001, S. 36 f).

3.7 The Bibtex-XML-HTML Project

<http://www.authopilot.com/xml/index.html> (Zugriff am 3.10.2002)

Michael Auth und Frank Richter vom Fachbereich für Linguistik der Universität Tübingen haben dieses Projekt ins Leben gerufen. Die Entwicklung des Konverters basiert auf den Vorarbeiten, die in der Diplomarbeit "BibTeXML" von Brenno Lurati und Luca Previtali (Lurati 2001) geleistet wurden.

Man hat einen Konverter entwickelt, mit dem es möglich ist, aus einer Bibtex-Datei eine XML-Datei zu generieren. In einem weiteren Schritt werden aus der XML-Datei mehrere unterschiedliche HTML-Dokumente erstellt.

Eines dieser HTML-Dokumente bildet eine Gesamtübersicht für alle Quellenangaben aus der Bibtex-Datei. Sie werden im Cambridge-Anzeigestil formatiert. Dieser HTML-Datei können pro Quellenangabe noch zusätzliche Informationen angehängt werden, wie z. B. die zugehörige URL oder eine PDF-Datei.

Weitere HTML-Dokumente enthalten jeweils eine einzige Quellenangabe. Die Anzeige erfolgt in der Formatierung des Bibtex-Eintrages, um ihn durch Kopieren und Einfügen aus der Zwischenablage ("copy and paste") in eine andere Bibtex-Datei einzufügen.

Zusätzlich müssen die Einträge der Bibtex-Datei vor der Konvertierung thematischen Kategorien zugeordnet werden. Nach diesen Kategorien werden weitere HTML-Dateien erstellt. Je mehr Kategorien festgelegt wurden, desto mehr Dateien existieren. Sie bilden einen Übersichtskatalog über die der Kategorie zugeteilten Bibtex-Einträge.

Um eine Übersicht über alle vorhandenen Kategorien zu bereitzustellen, werden auch sie in einer weiteren HTML-Datei aufgelistet.

Sind die HTML-Dateien einmal erstellt, sind sie statisch und können nicht mehr benutzerspezifisch geändert werden. Eine Sortierung der Quellenangaben kann nur vor der Umwandlung zu HTML im Bibtex-Format erfolgen.

In der folgenden Grafik von der Homepage des Projekts wird der Zusammenhang der verschiedenen Formate dargestellt (Bibtex-XML-HTML Project 2002):

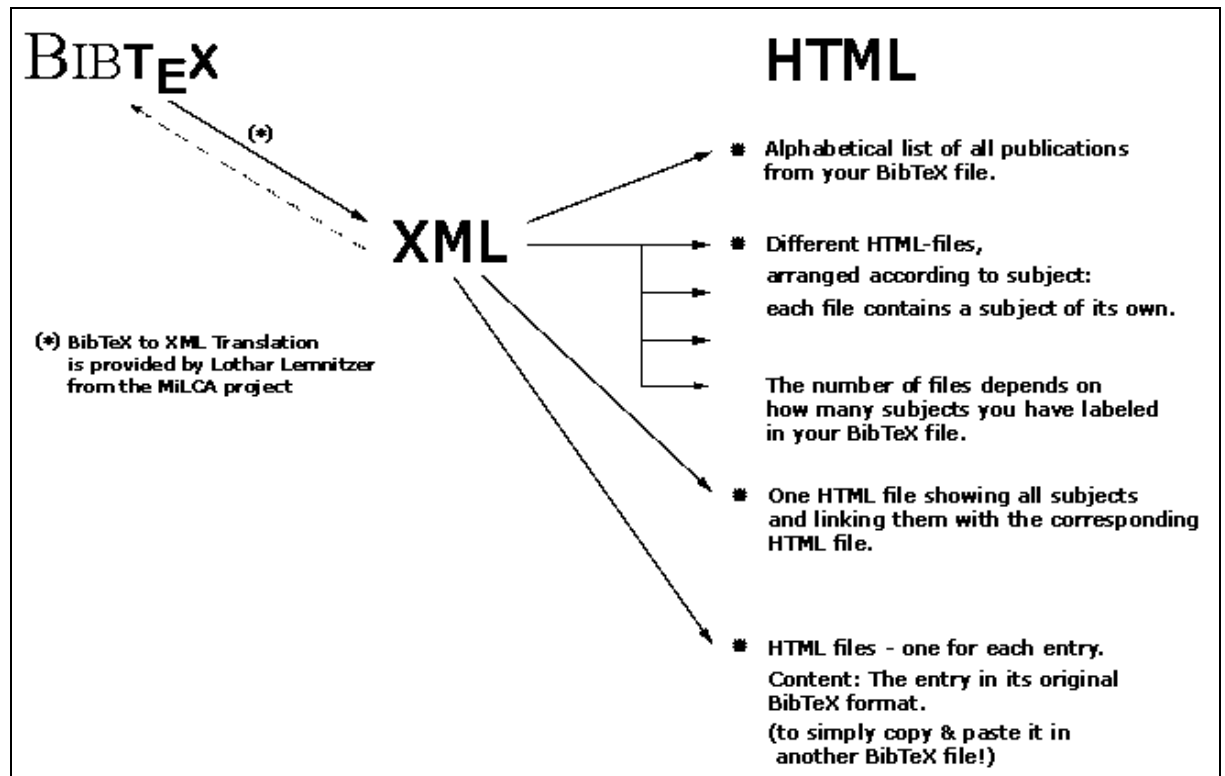


Abbildung 9: Bibtex-XML-HTML Project: Übersichtsdiagramm der erstellten HTML-Seiten

(Grafik entnommen aus Bibtex-XML-HTML Project 2002)

4 Zu unterstützende Darstellungsformate für Bibliographien

4.1 Übersicht

Die Aufgabenstellung dieser Diplomarbeit ist es, eine Bibliographiedatei, die im Bib-TeX-Format vorliegt, mit einem Webbrowser anzuzeigen und durchsuchbar zu machen. Die Ergebnisse sollen mit verschiedenen Bibliographiestilen ausgegeben und auf verschiedene Weise sortiert werden können. Die Anwenderoberfläche zur Suche bildet ein Formular, das in PHP programmiert ist und das dem Anwender verschiedene Eingabeoptionen bietet. Eingegeben werden können Autorname, Titel des Werks und Erscheinungsjahr. Zusätzlich soll eine Auswahloption ermöglichen, bei der Darstellung des Suchergebnisses nur eine bestimmte Art der Quellenangabe anzuzeigen. So lässt sich eine Auswahl treffen, dass die Liste zum Beispiel nur Zeitschriftenartikel enthalten soll. Neben einer Sortierfunktion sollen zusätzlich folgende Anzeigestile der Ergebnisliste dargestellt werden:

- DIN 1505 Teil 2
- APA (American Psychological Association)
- Harvard

Die unterschiedlichen Anzeigestile ermöglichen es dem Anwender, die Ergebnisdaten auf einfache Weise in eine eigene Bibliographie, z. B. in einem Worddokument durch Kopieren und Einfügen aus der Zwischenablage („copy and paste“) zu übernehmen, ohne die Formatierung der eingefügten Daten zusätzlich anpassen zu müssen.

4.2 Anzeigestile

Den Anzeigestilen für DIN 1505, APA und Harvard ist jeweils ein eigenes Stylesheet zugeordnet. Jedes XSL-Stylesheet beinhaltet wiederum Teile zur Darstellung bestimmter Arten bibliographischer Elemente. Dazu gehören nach Lorenzen (Lorenzen 1997, S. 3) unter anderem: Monographien, Sammelwerke, Tagungsschriften, Zeitschriftenbände, Forschungsberichte, Firmenschriften, Hochschulschriften, Loseblattausgaben, Normen, Patente, Ausstellungskataloge, Filme, elektronische Dokumente, Mündliche Äußerungen, Zeitschriftenaufsätze, Zeitungsartikel, Einzelbeiträge in einem Sammelband, Kapitel einer Monographie, Vorträge auf einer Konferenz.

Die Einteilung der Formate nach deren Anzeigestil in verschiedenen XSL-Stylesheets ist bedingt durch die speziellen Anforderungen der Anzeige für jeden einzelnen Stil. Andernfalls kann die Anzeige nicht homogen dargestellt werden.

Um den Umfang dieser Diplomarbeit zu beschränken werden hier nur die geläufigsten Formate beschrieben und implementiert. Die Stylesheets können ohne großen Aufwand erweitert werden. Daher stellt das nachträgliche Einfügen von benötigten Formaten keine Schwierigkeit dar.

In dieser Diplomarbeit verwendete bibliographische Elemente sind:

- Monographien
- Tagungsschriften
- Forschungsberichte
- Firmenschriften
- Hochschulschriften
- Zeitschriftenaufsätze
- Einzelbeiträge in einem Sammelband
- Vorträge auf einer Konferenz
- unveröffentlichte Werke

Außerdem ist es noch möglich, weitere Formate einzufügen, z. B.

- Sammelwerke
- Zeitschriftenbände
- Ausstellungskataloge
- Filme
- Elektronische Dokumente
- Zeitungsartikel

Die üblichen Elemente einer Quellenangabe für gedruckte Werke sind Autorname(n) oder Name einer Körperschaft, Werktitel, Verlagsort, Verlag, Erscheinungsjahr und ggf. Seitenangaben. Andere Quellen, wie z. B. elektronische Dokumente von Internetseiten benötigen andere Angaben. Z. B. kommen statt Angaben zu einem Verlag Angaben der zugehörigen URL vor. Wenn noch ein Format für elektronische Dokumente angelegt wird, wäre die Einbindung von Hyperlinks zu den angegebenen URLs in der Ergebnisliste wünschenswert.

4.2.1 Anzeigestil nach DIN 1505 Teil 2

DIN 1505 unterscheidet bei der Ansetzung zwischen selbständig und unselbständig erschienenen Einheiten. Die einzelnen Elemente der Quellenangaben werden durch Trennzeichen wie Komma, Semikolon oder Doppelpunkt getrennt und durch typographische Hervorhebungen kenntlich gemacht.

Da jede mögliche Art von bibliographischer Einheit geringfügig unterschiedliche Ansetzungsregeln besitzt, möchte ich hier nur beispielhaft die wichtigsten herausgreifen.

Bibliographisch selbständig erschienene Einheiten

Die Ansetzungsregeln für Monographien u. ä. hat Lorenzen in eine Tabelle gefasst, die hier beispielhaft dargestellt wird (geringfügige Änderung bei der Darstellung: Die Zeichen für Leerzeichen werden hier mit „_“ dargestellt):

Tabelle 1: nach Lorenzen 1997, S. 8. : Monographien, Sammelwerke, Forschungsberichte usw.

Monographien	
davor	bibliographische Elemente des Zitats
(_:_)	1. Verfasserangabe(n)
(_:_)	2. Herausgeber [dahinter: „(Hrsg.)“]
:	3. wichtige Mitarbeiter, Bearbeiter [dahinter: „(Mitarb.)“]
(_:_)	4. Körperschaftl. Urheber [dahinter: „(Hrsg.)“]
:_	5. <i>Sachtitel</i> [kursiv gesetzt]
:	6. <i>Zusatz zum Sachtitel</i> [kursiv gesetzt]
/	7. weniger wichtige beteil. Pers. [dahinter: „(Bearb.)“]
._	8. Bandangabe
,_	9. Teilbandzählung
:	10. <i>Sachtitel d. (Teil-)Bandes</i> [kursiv]
._	11. Auflage (Ausgabebezeichnung)
._	12. erster Erscheinungsort
:	13. erster Verlag
,_	14. Erscheinungsjahr(e)
_	15. 1. Gesamttitel mit Zählung [in rund. Klammern]
_	16. 2. Gesamttitel mit Zählung [in rund. Klammern]
.__-	17. Ergänzende Angaben [z.B. Umfangsangabe]
.__-ISBN	18. ISBN-Angabe

(Tabelle wurde geringfügig abgewandelt: Leerzeichen durch „_“ ersetzt)

Diese Darstellungsform gilt für bibliographisch selbständig erschienene Einheiten wie z. B. Monographien, Sammelwerke, Forschungsberichte, Reports, Firmenschriften

oder Karten. Bei Band- und Teilbandangaben steht zuerst die Angabe des „hierarchisch übergeordneten, dann der Titel des Teils“ (Lorenzen 1997, S. 20).

Bibliographisch unselbständig erschienene Einheiten

Für unselbständig erschienene Einheiten gelten differenziertere Regeln als für selbständig erschienene. Zeitschriften- oder Zeitungsaufsätze werden anders angesetzt als z. B. Beiträge in Sammelwerken oder Vorträge auf einer Konferenz. Oft gelten für unselbständig erschienene Einheiten Ansetzungsregeln, die Teile der Ansetzung für Monographien enthalten, denen jedoch zusätzliche Ergänzungen oder Angaben vorangestellt oder angehängt werden.

Beispielhaft soll die Tabelle „Aufsätze in Zeitschriften und Zeitungen“ von Lorenzen (Lorenzen 1997, S. 13) die Ansetzung bibliographisch unselbständig erschienener Einheiten zeigen. Der erste Teil der Ansetzung enthält alle Informationen zum Aufsatz, der zweite Teil enthält die Angaben zur Zeitschrift, in der der Aufsatz erschienen ist. Verbunden werden die beiden Teile durch „In:“.

Tabelle 2: nach Lorenzen 1997, S. 13: Aufsätze in Zeitschriften und Zeitungen

Aufsätze in Zeitschriften u. Zeitungen	
davor	bibliographische Elemente des Zitats
(;_)	1. Verfasserangabe(n) [<i>alle!</i>]
:_	2. Sachtitel
:_	3. Zusatz zum Sachtitel
:_	4. Angabe des Teils [<i>wenn in Forts. veröff.</i>]
:_	5. Gesamttitelangabe [<i>in runden Klammern</i>]
._In:_	6. <i>Titel der Zeitschrift / Zeitung</i> [<i>kursiv!</i>]
,'_	7. <i>Unterreihe der Zeitschrift / Zeitung</i>
_	8. Bandzählung der Zeitschrift
_	9. Erscheinungsjahr [<i>in runden Klammern</i>]
._Nr._	10. Heftnummer
._S._	11. Seitenzählung [<i>erste - letzte Zahl</i>]

(Tabelle wurde geringfügig abgewandelt: Leerzeichen durch „_“ ersetzt)

Bei anderen unselbständig erschienenen Einheiten erfolgt die Ansetzung nach einem ähnlichen Schema, jedoch mit unterschiedlichen Einzelangaben.

Trotz aller Unterschiede in der Ansetzung der einzelnen Einheiten gibt es auch Gemeinsamkeiten, die grundsätzlich bei jeder Einheit auftauchen. Nach DIN 1505 Teil 2 werden beispielsweise Verfasserangaben grundsätzlich in Kapitälchen gesetzt. Der Nachname des Autors steht vor dem mit einem Komma abgetrennten Vornamen. Bei mehreren Autoren trennt ein Semikolon die einzelnen Namen, es folgt ein Doppelpunkt und der Titel der Einheit mit einem Punkt (Lorenzen 1997, S. 6).

Als Beispiel:

HERPERS, FRANZ-JOSEF ; SEBESTYEN, THOMAS J.: *Das Einsteigerseminar XSL*.

Auch Körperschaften, die als Verfasser anzusetzen sind, werden in Kapitälchen gesetzt, eventuell auch mit zusätzlichen Angaben wie „(Hrsg.)“ für Herausgeber oder „(Veranst.)“ für veranstaltende Körperschaft (Lorenzen 1997, S. 15).

Literaturverzeichnis

Ein Literaturverzeichnis nach DIN 1505 Teil 2 enthält zusätzlich zu den bibliographischen Quellenangaben noch eine Einordnungsformel. Nach dieser Formel werden die zitierten Dokumente alphanumerisch geordnet (Lorenzen 1997, S. 20). Ein Beispiel für eine Quellenangabe mit vorangestellter, fettgedruckter Einordnungsformel ist bei Lorenzen (Lorenzen 1997, S. 21) zu finden:

Franke 1978 FRANKE, HERBERT W. ; SCHRAUD, PETER: Sachliteratur zur Technik. In: RADLER, R. (Hrsg.): *Die deutschsprachige Sachliteratur*. München : Kindler, 1978 (Kindlers Literaturgeschichte der Gegenwart), S. 654-676

4.2.2 Anzeigestil nach APA (American Psychological Association)

„APA“ ist eine der am häufigsten gebrauchten Vorschriften zur Erstellung wissenschaftlicher Arbeiten im englischen Sprachraum und enthält auch Anweisungen zur Zitierweise und zu Literaturangaben. Die Vorschriften erscheinen in dem Handbuch „Publication Manual of the Psychological Association“ (American Psychological Association 2001) und werden regelmäßig überarbeitet.

Dieser Stil unterscheidet ähnlich wie der nach DIN 1505 nach selbständig und unselbständig erschienenen Einheiten, wie bereits im vorangegangenen Kapitel beispielhaft gezeigt wurde.

Zusätzlich zu diesen Ansetzungsformen, die besonders für Druckwerke gelten, existieren seit der 5. Ausgabe des Publication Manual of the Psychological Association (2001) weitere Regeln zur Ansetzung elektronischer Quellen, die wieder in selbständige und unselbständige eingeteilt werden. Es gelten nun Ansetzungsregeln, z. B. für Quellen aus Newsgroups, E-Mails, oder „Real-Time“-Kommunikation wie aus Chatrooms (OWL 2001).

Bei der Liste der Literaturangaben gilt laut OWL (OWL 2001) für jede einzelne Quelle ein hängender Einzug für alle der ersten Zeile folgenden Zeilen, Invertierung des Autornamens zur Form „Nachname, Vorname“ sowie Kursivsatz für Quellentitel.

Die Sortierung ist streng geregelt nach Verfasseramen, anschließend nach dem aufsteigenden Datum der Veröffentlichung. Es existiert auch eine Regelung für den Sonderfall, dass ein Verfasser einmal als einziger Verfasser eines Werkes in der Liste auftaucht und zusätzlich als erster Autor einer Quelle mit mehreren Verfassern. Hier gilt: Der Eintrag für den Einzelverfasser wird zuerst angegeben.

Falls einmal kein Verfasser identifiziert werden kann, wird der Titel der Quelle in der Sortierung wie ein Verfassername verwendet.

Besonders hervorzuheben sind für deutschsprachige Leser ins Auge fallende Unterschiede zu DIN 1505. Ein Beispiel dafür ist die Ansetzung eines zweiten Verfasser Namens, der, von einem Komma vom ersten Verfasser getrennt, mit einem „Kaufmanns-Und“ (Ampersand, &) angeschlossen wird. Sämtliche Vornamen eines oder mehrerer Verfasser werden nur mit dem ersten Buchstaben angegeben, gefolgt von einem Punkt.

Ein weiterer maßgeblicher Unterschied ist, dass das Erscheinungsdatum bei allen Einheiten in Klammern direkt der Autorenangabe folgt. Eventuell wird bei unselbständig erschienenen Einheiten wie Artikel aus Zeitschriften oder Zeitungen zusätzlich neben der Angabe des Jahres- noch das Monats- und Tagesdatum angegeben. Es entsteht eine ähnliche Form wie bei der Einordnungsformel beim DIN-Stil. Der Unterschied ist jedoch, dass beim APA-Stil keine weitere Datumsangabe folgt, während beim DIN-Stil noch einmal das Erscheinungsjahr nach den Angaben zum Verlag oder dem übergeordneten Werk auftauchen muss.

Folgende Beispiele stammen von der Website des OWL der Purdue University (OWL 2001):

Beispiel für eine Monographie mit zwei Autoren:

Calfee, R. C., & Valencia, R. R. (1991). *APA guide to preparing manuscripts for journal publication*. Washington, DC: American Psychological Association.

Beispiel für einen Artikel aus einer Zeitschrift:

Henry, W. A., III . (1990, April 9). Making the grade in today's schools. *Time*, 135, 28-31.

Eine Quellenangabe wird grundsätzlich mit einem Punkt abgeschlossen.

4.2.3 Anzeigestil nach Harvard

Für den Harvard-Stil gilt nach der Cybrary der University of Queensland (Cybrary 2001):

- Die Angaben zum Erscheinungsdatum erfolgen nicht in Klammern.
- Die Verfasseramen sind vom Erscheinungsdatum nicht mit einem Trennzeichen getrennt.
- Der Titel und der Untertitel der Quelle können unterstrichen oder kursiv gekennzeichnet werden.
- Falls die Monographie nicht in der ersten Auflage erschienen ist, muss sie angegeben werden.
- Herausgeber werden mit „ed.“ bzw. „eds“ abgekürzt.
- Wie beim APA-Stil erfolgt eine Einrückung ab der zweiten Zeile der Quellenangabe.

- Das Ende eines Quelleneintrags wird durch einen Punkt gekennzeichnet.
- Der letzte von mehreren Verfassern wird durch ein „&“-Zeichen angeschlossen.

Ein Beispiel für mehrere Autoren eines Buches der Cybrary (Cybrary 2001):

Cheek, J., Doskatsch, I., Hill, P. & Walsh, L. 1995, *Finding Out* :

Information Literacy For the 21st century, MacMillan Education Australia, South Melbourne.

Falls keine Verfasserangabe feststellbar ist, wird die Quelle nach dem ersten Hauptwort des Titels in der Literaturliste einsortiert.

Mit unselbständigen Einheiten, wie Artikeln aus Zeitschriften, wird nach einem anderen Schema verfahren. Der Titel des Artikels wird in einfache Anführungszeichen eingefasst (') und nicht kursiv gekennzeichnet. Es folgt ein Komma und anschließend kursiv der Titel der Zeitschrift oder des übergeordneten Werkes. Bandzählung, Teilnummer oder Ausgabennummer und Seitenanzahl werden von Kommas getrennt angegeben.

Beispiel für einen Zeitschriftenartikel (Cybrary 2001):

Huffman, L. M. 1996, 'Processing whey protein for use as a food ingredient', *Food Technology*, vol. 50, no. 2, pp. 49-52

Bei Konferenzbeiträgen wird zusätzlich der Ort hinzugefügt.

Beispiel für einen Beitrag einer Konferenz (Cybrary 2001):

Bohrer, S., Zielke, T. & Freiburg, V. 1995, 'Integrated obstacle detection framework for intelligent cruise control on motorways', *IEEE Intelligent Vehicles Symposium*, Detroit, MI, Piscataway, pp. 276-281.

Non-Book-Einheiten werden wie Monographien behandelt, ihnen wird jedoch nach dem Titel in Klammern die Art des Werks beigefügt, z. B. „videorecording“, „cd-rom“.

Beispiel für eine Non-Book-Einheit (Cybrary 2001):

Get the Facts (And Get Them Organised) (videorecording) 1990,
Williamstown, Vic., Appleseed Productions.

Internet-Ressourcen werden ähnlich behandelt.

Beispiel für die Angabe einer Internet-Quelle (Cybrary 2001):

Weibel, S. 1995, 'Metadata : the foundations of resource description',

D-lib Magazine, [Online] Available at:
<http://www.dlib.org/dlib/July95/07weibel.html>

In diesem Fall handelt es sich um eine Quelle, die sowohl im Internet, als auch in Printform erschienen ist. Die Angabe bezieht sich auf die Internet-Version (Angabe von [Online]). Bei Internetquellen steht oft kein Punkt als Abschlusszeichen, wenn der letzte Teil der Quellenangabe eine URL-Adresse ist.

5 Konzeption der Bibliographieanzeige

5.1 Übersicht

Im Rahmen dieser Diplomarbeit soll eine webbasierte Anzeige und Auswahlfunktion für Bibliographien erstellt werden. Die Anzeige soll die folgenden Anforderungen unterstützen:

- Im Internet benutzbar
- Suchfunktion
- Sortierfunktion
- Auswahl der Anzeige nach den Stilen Harvard APA oder DIN 1505

5.2 Anzeige im Internet

Für die Anzeige im Internet müssen besondere Aufgaben erfüllt werden. Es muss eine Formularfunktion geben, die im Internet zugänglich und über einen Webbrowser ansprechbar ist. Die Anzeige der Bibliographie soll direkt im Webbrowser erfolgen. Das bedeutet, dass spezielle Programmbefehle nötig sind, die einen XSL-Prozessor unter besonderen Bedingungen (Eingabevariablen, Stil der Ausgabe) starten.

Dies soll mit einem Formular erreicht werden, das in PHP programmiert ist. Mit dieser Skriptsprache lassen sich eingegebene Werte überprüfen. Es existiert auch eine spezielle Befehlsroutine, mit der man einen XSL-Prozessor starten kann. Durch Teile des PHP-Codes kann die Ausgabe des Prozessors angezeigt werden.

Die Anzeige soll dynamisch sein, d. h. jedes Mal, wenn Einträge im Formular abgeschickt werden, werden zuerst alle erforderlichen Programmroutinen des PHP-Skripts und anschließend der XSL-Prozessor gestartet. Der Prozessor generiert die Ausgabe, die vom PHP-Skript angezeigt wird, bei jedem Klick auf den Absende-Button neu.

5.3 Suchfunktion

Um eine Suche durchzuführen, sollen in das Formular Werte eingegeben werden können, die die Anzeige auf bestimmte Quelleneinträge beschränken.

Gesucht werden soll nach:

- Autorname
- Titel des Werks
- Erscheinungsjahr

- Art des Eintrags

Für die technische Umsetzung werden das PHP-Skript und besondere XSL-Transformationsroutinen benötigt, die in Kombination eine einschränkende Suche möglich machen. Dadurch sollen Eingabewerte für z. B. Titel und Jahr eine AND-Verknüpfung bilden. Dadurch wird das Suchergebnis auf Quellenangaben beschränkt, die beide Suchwerte beinhalten.

5.4 Sortierfunktion

Die Anzeige soll sortierbar sein

- nach dem Autornamen
- nach dem Erscheinungsjahr

Der Anwender kann auswählen, ob die Ausgabe aufsteigend ("A-Z") oder absteigend ("Z-A") sortiert werden soll.

5.5 Auswahl des Anzeigestils

Jede Ausgabe kann auf unterschiedliche Weise formatiert werden. Für den Anwender besteht die Auswahl zwischen einem Anzeigestil für die Quellenangaben nach DIN 1505 Teil 2, APA oder Harvard.

6 Benutzung aus Sicht des Anwenders

6.1 Übersicht

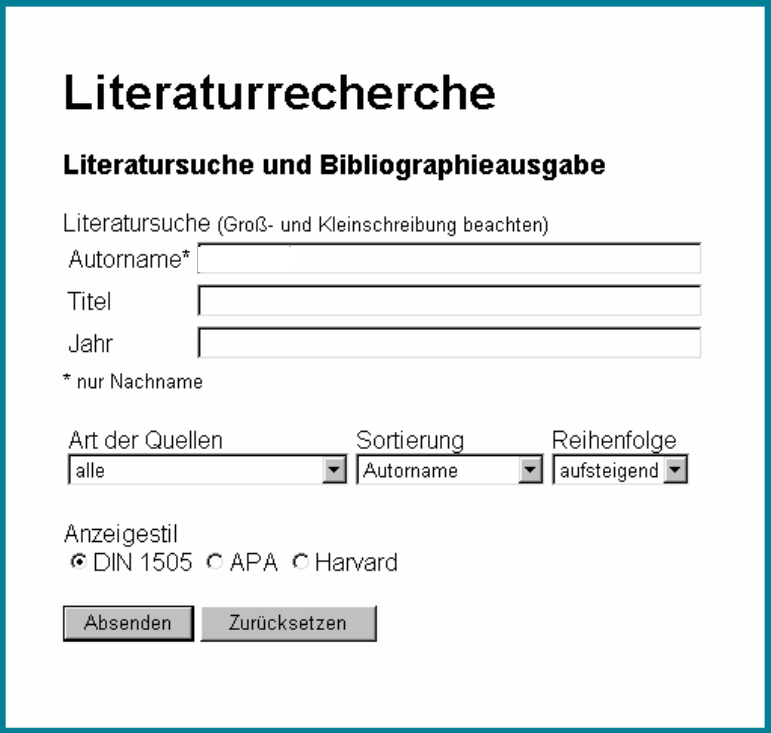
Nach den Anforderungen, die im letzten Kapitel beschrieben wurden, wurde eine Anzeige gestaltet. Ihre Benutzung aus Anwendersicht wird in diesem Kapitel gezeigt. Beschrieben wird die Benutzung der Such-, Sortier- und Anzeigefunktionen.

Bis auf weiteres ist das funktionsfähige Anzeigeformular unter der URL

<http://mars.iuk.hdm-stuttgart.de/projekte/kersten/Recherche.php> erreichbar.

6.2 Suchformular

Mit dem Aufruf der Formulardatei im Browser öffnet sich das Suchformular.



Literaturrecherche

Literatursuche und Bibliographieausgabe

Literatursuche (Groß- und Kleinschreibung beachten)

Autornamen*

Titel

Jahr

* nur Nachname

Art der Quellen Sortierung Reihenfolge

Anzeigestil
☒ DIN 1505 ☐ APA ☐ Harvard

Abbildung 10: Suchformular

Es enthält Eingabefelder für den Autornamen, den gesuchten Titel und das Erscheinungsjahr. Bei der Eingabe ist es wichtig, Groß- und Kleinschreibung zu beachten. Die Suche erfolgt in den Inhalten der XML-Datei. Das bedeutet, es können nur Ergebnisse ausgegeben werden, mit denen eine exakte Übereinstimmung besteht. Eine Trunkierung mit " * " oder anderen Wildcards sowie Bool'sche Ausdrücke (z. B. AND, OR) lie-

fern keine Ergebnisse. Allerdings lassen sich Wortfragmente eingeben, die eine rudimentäre Trunkierung möglich machen. So ergibt der Eintrag "frag" im Suchfeld "Titel" neben Ergebnissen, die das Wort "fragen" enthalten, auch z. B. "Anfrage" oder "gefragte", die dieses Wortfragment an beliebiger Stelle beinhalten. Das Wort "Frage" wird nicht gefunden, denn die Suche ist case-sensitive, d. h. Groß- und Kleinschreibung wird berücksichtigt.

Eine Auswahl an Optionsfeldern ermöglicht die Suche nach verschiedenen Arten von Literatur:

- alle
- Buch
- Aufsatz in einem Buch
- Artikel in einer Zeitschrift
- Tagungsbericht
- Teil einer Aufsatzsammlung
- Techreport
- Diplomarbeit
- Unveröffentlicht

Eine Sortierung der Ergebnisse ist nach Autornamen und Erscheinungsjahr in aufsteigender und absteigender Reihenfolge möglich.

Die Anzeige kann nach den folgenden Anzeigestilen ausgegeben werden:

- DIN 1505
- Harvard
- APA

Mit dem Anklicken des Buttons "Suche" wird das zugrundeliegende PHP-Skript gestartet. Es wertet aus, welche Optionsfelder und Suchbegriffe angegeben wurden und startet den XSL-Prozessor mit diesen Eingaben. Der Prozessor gibt die Ergebnisdaten an das PHP-Formular zurück, das eine Ergebnisliste anzeigt. Wurde kein Treffer gefunden, enthält die Liste keinen Inhalt.

6.3 Suchbeispiele

6.3.1 Suche nach Autorname

Zur Suche nach allen Werken eines Autors ist folgendes Vorgehen nötig:

- Den gesuchten Nachnamen vollständig in das Feld Autorname eintragen.
- Quellenart "alle" auswählen

- Einen Anzeigestil auswählen
- Die Suche mit Klick auf den "Suche"-Button starten

Abbildung 11: Suche nach Autor

Das Suchergebnis wird ausgegeben. Passende Einträge werden im gewünschten Anzeigestil angezeigt. Das Suchergebnis ist alphabetisch sortiert nach dem ersten Autornamen und innerhalb davon nach Erscheinungsjahr.

Abbildung 12: Suche nach Autor - Suchergebnis (oberer Teil)

Bibliographie nach Harvard-Stil

Alexander, C. 1964, *Notes on the Synthesis of Form*. Harvard University Press.
 Alexander, C. 1971, 'A Refutation of Design Methodology (Interview with Max Jacobson)', *Architectural Design* - December
 Alexander, C. and Silverstein, M. and Angel, S. 1975, *The Oregon Experiment*. Oxford University Press.
 Alexander, C. and Ishikawa, S. and Silverstein, M. 1977, *A Pattern Language*. Oxford University Press.
 Alexander, C. 1979, *A Timeless Way of Building*. Oxford University Press.
 Alexander, C. 1981, *The Linz Cafe*. Oxford University Press.
 Alexander, C. 1985, *The Production of Houses*. Oxford University Press.
 Alexander, C. 1987, *A New Theory of Urban Design*. Oxford University Press.
 Alexander, C. 1991, 'Perspectives: Manifesto 1991', *Progressive Architecture* - July

 Anfang  neue Suche

Abbildung 13: Suche nach Autor - Suchergebnis (unterer Teil)

6.3.2 Verknüpfung von Suchbegriffen

Eine Verknüpfung von gesuchten Begriffen ist nur möglich, wenn sich die Suchbegriffe nicht überschneiden. So kann man eine Suche ausführen, die nach einem Autornamen und einem Titel gleichzeitig einschränkt. Für ein sinnvolles Ergebnis sollten jedoch nicht zwei Autornamen gleichzeitig angegeben werden.

Dieses Suchbeispiel liefert eine Quelle zurück, bei der sowohl Autor und Erscheinungsjahr mit den eingegebenen Werten übereinstimmen. "Alexander" und "1991" bringt als Ergebnisanzeige:

Literaturrecherche

Sie haben folgende Auswahl getroffen:

Autorenname: Alexander

Titel:


Erscheinungsjahr: 1991

Literaturart:

Sortierung: Autornamen

Reihenfolge: aufsteigend

Anzeigestil: Harvard

 neue Suche

Bibliographie nach Harvard-Stil

Alexander, C. 1991, 'Perspectives: Manifesto 1991', *Progressive Architecture* - July


Abbildung 14: Suchergebnis für Autor und Jahr

Eine Verknüpfung ist auch mit der Art der Quellenangabe möglich. Die gleichzeitige Sucheingabe mehrerer Begriffe, z. B. von Autor, Titel und Art der Quelle, liefert bei einer Bibliographie mit wenigen Einträgen jedoch meist keine oder geringe Trefferquoten.

Literaturrecherche

Sie haben folgende Auswahl getroffen:

Autorenname: Coplien
Titel:
Erscheinungsjahr: 1991
Literaturart: book
Sortierung: Autorname
Reihenfolge: aufsteigend
Anzeigestil: Harvard

 neue Suche

Bibliographie nach Harvard-Stil

Coplien, J. 1991, *Advanced C++: Programming Styles and Idioms*. Addison-Wesley.

Abbildung 15: Suchergebnis für Autor, Jahr und Literaturart

6.4 Anzeige der gesamten Bibliographie

Um alle Einträge der Bibliographiedatei auszugeben, genügt es, einen Anzeigestil auszuwählen. Wenn die anderen Suchfelder nicht ausgefüllt sind, bzw. bei Literaturart "alle" ausgewählt ist, wird die gesamte Bibliographieliste in dem ausgewählten Stil angezeigt.

Literaturrecherche

Literatursuche und Bibliographieausgabe

Literatursuche (Groß- und Kleinschreibung beachten)

Autoname*

Titel

Jahr

* nur Nachname

Art der Quellen Sortierung Reihenfolge

Anzeigestil

☒ DIN 1505 ☐ APA ☐ Harvard

Abbildung 16: Suchformular für die Ausgabe der gesamten Bibliographie

Literaturrecherche

Sie haben folgende Auswahl getroffen:

Autorenname:

Titel:

Erscheinungsjahr:

Literaturart:

Sortierung: Autornamen

Reihenfolge: aufsteigend

Anzeigestil: Din

 neue Suche

Bibliographie nach DIN 1505-Stil

Adelson 1981 ADELSON, B.: Problem Solving and the Development of Abstract Categories in Programming Languages. In: *Memory and Cognition* 9 (1981), S. 422-433

Akin 1986 AKIN, OMER: *Psychology of Architectural Design*. Pion Limited, 1986

Alexander 1964 ALEXANDER, C.: *Notes on the Synthesis of Form*. Harvard University Press, 1964

Alexander 1971 ALEXANDER, C.: A Refutation of Design Methodology (Interview with Max Jacobson). In: *Architectural Design* (1971) - December

Alexander 1975 ALEXANDER, C.; SILVERSTEIN, M.; ANGEL, S.: *The Oregon Experiment*. Oxford University Press,

Abbildung 17: Ausgabe der gesamten Bibliographie – oberer Teil

Shaw 1991 SHAW, M.: Heterogeneous Design Paradigms for Software Architecture. In: *Proceeding of the Sixth International Workshop on Software Specification and Design* - October 25-26. Como, Italy, 1991 (Software Engineering Notes), S. 158-165

Simon 1981 SIMON, H.: *The Sciences of the Artificial*. MIT Press, 1981

Soloway 1983 SOLOWAY, E.; EHRLICH, K.: Empirical Studies of Programming Knowledge. In: *IEEE Transactions on Software Engineering* 10, Nr. 4, S. 595-609

Soloway 1983 SOLOWAY, E.; BONAR, J.; EHRLICH, K.: Cognitive Strategies and Looping Constructs: An Empirical Study. In: *Communications of the ACM* (1983), S. 853-860

Soukup 1994 SOUKUP, JIRI: *Taming C++: Pattern Classes and Persistence for Large Projects*. Addison-Wesley, 1994

Spaleck 2001 SPALECK, SEBASTIAN: *XML-basierte Generierung individueller Terminpläne*. Stuttgart, Hochschule der Medien, Diplomarbeit, 2001

Van Dijk 1983 VAN DIJK, T.; KINTSCH, W.: *Strategies of Discourse Comprehension*. New York: Academic Press, 1983

Wiedenbeck 1986 WIEDENBECK, S.; SCHOLTZ, J.: Beacons: a Knowledge Structure in Program Comprehension. In: G. SALVENDY & M. SMITH: *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*. Amsterdam: Elsevier, S. 82-87

Wiedenbeck 1986 WIEDENBECK, S.: Processes in Computer Comprehension. . Norwood, NJ: Ablex, 1986, S. 48-57

Williams 1990 WILLIAMS, JOSEPH: *Style*. The University of Chicago Press, 1990

Winograd 1986 WINOGRAD, T.; FLORES, F.: *Understanding Computers and Cognition: A New Foundation for Design*. Addison-Wesley, 1986

Woodside 1993 WOODSIDE, C. M.; FRANKS, R. G.: Alternative Software Architectures for Parallel Protocol Execution with Synchronous IPC. In: *IEEE/ACM Transactions on Networking* 1 (1993), Nr. 2 - April, S. 178-186

 Anfang  neue Suche

Abbildung 18: Ausgabe der gesamten Bibliographie - unterer Teil

6.5 Sortierung


Eine Sortierung ist möglich nach Autornamen oder Erscheinungsjahr. Zusätzlich kann noch eine auf- oder absteigende Reihenfolge gewählt werden.

In den folgenden Beispielen wurde eine Suche nach Teilen einer Aufsatzsammlung durchgeführt, die beim Erscheinungsjahr die Zahlenkombination "19" enthalten.

Sortierung nach Autor:

Sie haben folgende Auswahl getroffen:

Autorenname:
 Titel:
 Erscheinungsjahr: 19
 Literaturart: incollection
 Sortierung: Autornamen
 Reihenfolge: aufsteigend
 Anzeigestil: Harvard

 neue Suche

Bibliographie nach Harvard-Stil

Ehrlich, K. and Soloway, E. 1984, 'An Empirical Investigation of the Tacit Plan Knowledge in Programming' in *Human Factors and Computer Systems*, J.C. Thomas and M.L. Schneider, Norwood, NJ Ablex, pp. 113-133.


Littman, D. and Pinto, J. and Letovsky, S. and Soloway, E. 1986, 'Mental Models and Software Maintenance' in *Empirical Studies of Programmers: First Workshop*, E. Soloway & S. Iyengar, Ablex, pp. 80-98.

Wiedenbeck, S. 1986, 'Processes in Computer Comprehension' Norwood, NJ Ablex, pp. 48-57.

Abbildung 19: Sortierung aufsteigend nach Autor

Sie haben folgende Auswahl getroffen:

Autorenname:
 Titel:
 Erscheinungsjahr: 19
 Literaturart: incollection
 Sortierung: Jahr
 Reihenfolge: absteigend
 Anzeigestil: Harvard

 neue Suche

Bibliographie nach Harvard-Stil


Wiedenbeck, S. 1986, 'Processes in Computer Comprehension' Norwood, NJ Ablex, pp. 48-57.

Littman, D. and Pinto, J. and Letovsky, S. and Soloway, E. 1986, 'Mental Models and Software Maintenance' in *Empirical Studies of Programmers: First Workshop*, E. Soloway & S. Iyengar, Ablex, pp. 80-98.

Ehrlich, K. and Soloway, E. 1984, 'An Empirical Investigation of the Tacit Plan Knowledge in Programming' in *Human Factors and Computer Systems*, J.C. Thomas and M.L. Schneider, Norwood, NJ Ablex, pp. 113-133.

Abbildung 20: Sortierung absteigend nach Autor


Sortierung nach Jahr:

Sie haben folgende Auswahl getroffen:
Autorenname:
Titel:
Erscheinungsjahr: 19
Literaturart: incollection
Sortierung: Jahr
Reihenfolge: aufsteigend
Anzeigestil: Harvard
 neue Suche

Bibliographie nach Harvard-Stil

Ehrlich, K. and Soloway, E. 1984, 'An Empirical Investigation of the Tacit Plan Knowledge in Programming' in *Human Factors and Computer Systems*, J.C. Thomas and M.L. Schneider, Norwood, NJ Ablex, pp. 113-133.
Littman, D. and Pinto, J. and Letovsky, S. and Soloway, E. 1986, 'Mental Models and Software Maintenance' in *Empirical Studies of Programmers: First Workshop*, E. Soloway & S. Iyengar, Ablex, pp. 80-98.
Wiedenbeck, S. 1986, 'Processes in Computer Comprehension' Norwood, NJ Ablex, pp. 48-57.

Abbildung 21: Sortierung aufsteigend nach Jahr

Sie haben folgende Auswahl getroffen:
Autorenname:
Titel:
Erscheinungsjahr: 19
Literaturart: incollection
Sortierung: Jahr
Reihenfolge: absteigend
Anzeigestil: Harvard
 neue Suche

Bibliographie nach Harvard-Stil

Wiedenbeck, S. 1986, 'Processes in Computer Comprehension' Norwood, NJ Ablex, pp. 48-57.
Littman, D. and Pinto, J. and Letovsky, S. and Soloway, E. 1986, 'Mental Models and Software Maintenance' in *Empirical Studies of Programmers: First Workshop*, E. Soloway & S. Iyengar, Ablex, pp. 80-98.
Ehrlich, K. and Soloway, E. 1984, 'An Empirical Investigation of the Tacit Plan Knowledge in Programming' in *Human Factors and Computer Systems*, J.C. Thomas and M.L. Schneider, Norwood, NJ Ablex, pp. 113-133.

Abbildung 22: Sortierung absteigend nach Jahr

6.6 Verwendung in einer Worddatei

Das Suchformular ermöglicht eine Ausgabe, die der Benutzer nach seinen Anforderungen anpassen kann. Die Ergebnisliste oder Teile davon können kopiert und aus der Zwischenablage in ein Worddokument eingefügt werden ("copy and paste"). Die Formatierung wird in neuen Versionen von Winword (getestet ab Word 97) automatisch übernommen, so dass, bis auf die Einrückung, keine zusätzlichen Anpassungen mehr vorgenommen werden müssen.

6.7 Einbindung in eine Homepage

Um die Anwendungsmöglichkeiten der Bibliographieanzeige auszuweiten, ist es möglich, sie mit bestimmten voreingestellten Kriterien in einer anderen HTML-Seite einzubinden. So können beispielsweise Professoren ihre "persönliche" Bibliographieauswahl, deren Anzeigekriterien sie selbst bestimmen können, auf ihrer Homepage anzeigen.

Die Realisierung erfolgt über einen Link, der die Abfrageeinstellungen des Formulars beinhaltet. Wird auf den Link geklickt, erscheint die Ergebnisliste mit den vorab gewählten Auswahleinstellungen.

Die Einstellungen müssen einmal in das Suchformular eingetragen und abgesendet werden, das unter der URL

<http://mars.iuk.hdm-stuttgart.de/projekte/kersten/Recherche.php>

zu erreichen ist. Nach der Ausgabe des Suchergebnisses sieht man in der Adresszeile des Browsers eine veränderte URL, z. B:

```
http://mars.iuk.hdm-stuttgart.de/projekte/kersten/Recherche.php?lastname=
Coplien&title=&year=1991&kind=book&sortierung=Autornome&reihe=aufsteigend&
style=Harvard&bib=xml&sent=1
```

In dieser URL wurden durch das PHP-Skript die Variablen angefügt, die durch die Eingaben im Suchformular geändert wurden. Es genügt, diese verlängerte URL als Link in den HTML-Code der anderen Homepage einzufügen. Nach einem Klick auf den Link in der Professoren-Homepage öffnet sich die Ergebnis-Liste mit den voreingestellten Such- und Anzeigoptionen.

Nach dieser Beschreibung des Tools von der Anwenderseite wird im folgenden Kapitel ausgeführt, wie die Anzeige programmiertechnisch realisiert wurde.

7 Realisierung

7.1 Übersicht

Die Erfassung der Auswahl- und Anzeigekriterien des Anwenders erfolgt über ein Formular, das im Browser durch ein PHP-Skript erzeugt wird. Die Formatierungsvorschriften der verschiedenen Anzeigestile mit den Sortierungsfunktionen sind in je einem einzelnen XSL-Stylesheet vorgegeben, die auf dem Server, zusammen mit der PHP-Datei, gespeichert sind. Der XSL-Prozessor lädt sowohl das passende Stylesheet, als auch die angegebene XML-Datei, verarbeitet die Inhalte dieser Dateien und gibt das Ergebnis aus.

Die Ergebnisse können vom Prozessor beispielsweise als XML, HTML, PDF oder in anderen Formaten ausgegeben werden. Zur Anzeige der Suchergebnisse aus der Bibliographie-Datei in einem PHP-Skript ist das HTML-Format am besten geeignet. Je nachdem welcher Anzeigestil gewählt wurde, wird eine andere XSL-Datei zur Verarbeitung in den Prozessor geladen. Dies wird bestimmt von den Eingaben in den Formularfeldern des PHP-Skriptes.

7.2 Interne Abfolge der Programmteile

Der Anwender gibt die Suchdaten in die Formularfelder ein. Zu den Suchdaten gehören Anzeigestil, Sortierung und Suchbegriffe, wie z. B. der Autorname.

Das PHP-Skript prüft die Eingaben auf Gültigkeit. Anschließend startet es den XSL-Prozessor mit den Anweisungen, die durch die Eingaben definiert wurden. Die Übergabe von Variablen, die für das benötigte Stylesheet bestimmt sind, an den Prozessor erfolgt an dieser Stelle. PHP startet den Prozessor mit den Parametern für dasjenige XSL-Stylesheet, das nach der Eingabeprüfung für den Anzeigestil festgelegt wurde. Zusätzlich erhält der Prozessor beim Start eine Angabe, welche XML-Datei zu verarbeiten ist.

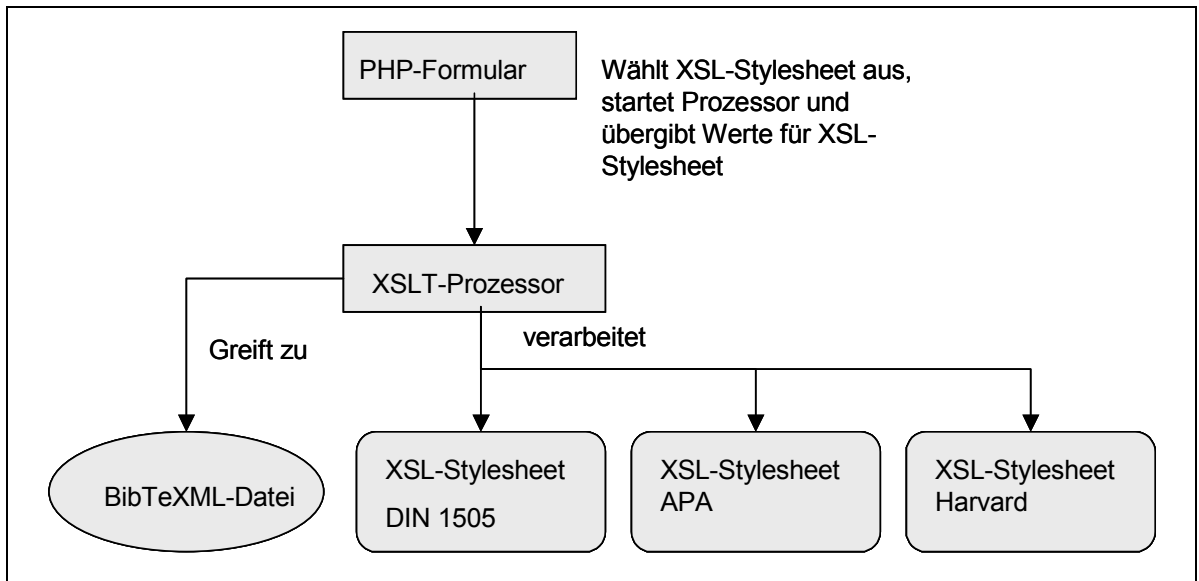


Abbildung 23: Auswertung der Formulareingaben

Da es in diesem Fall nur eine einzige XML-Datei gibt, die für die Verarbeitung zur Wahl steht, ist im Suchformular kein Eingabefeld angezeigt. Allerdings existiert ein verborgenes Formularfeld, um das spätere Hinzufügen von XML-Dateien zu unterstützen. In diesem Feld wird eine Variable vom PHP-Skript verarbeitet, die den Namen der XML-Datei für den Prozessoraufbau definiert. Für den Fall, dass weitere XML-Bibliographien durchsucht werden sollen, genügt es, das verborgene Feld sichtbar zu machen und weitere Auswahlmöglichkeiten hinzuzufügen.

Nachdem der XSL-Prozessor gestartet wurde und die Eingaben verarbeitet sind, gibt das PHP-Skript die Ergebnisliste im HTML-Format aus.

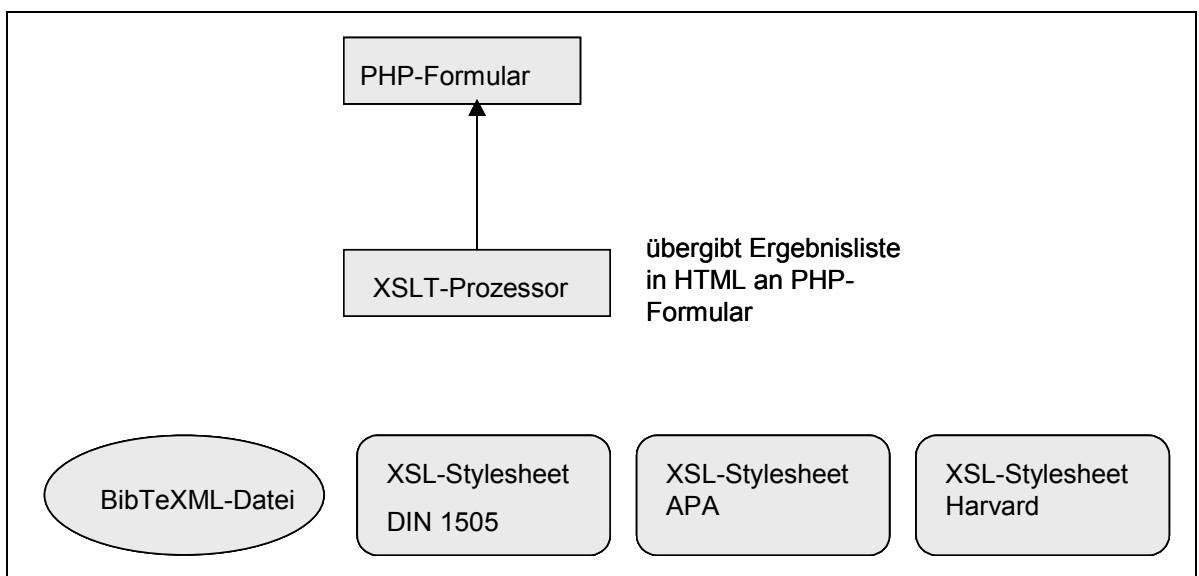


Abbildung 24: Ausgabe der Ergebnisliste

7.3 PHP-Formular

In den Formularfeldern des PHP-Skripts werden die Inhalte der Variablen festgelegt. Die Variable "\$lastname" enthält die Zeichenkette (String) aus dem Feld Auturname, "\$title" und "\$year" enthalten die Strings aus den Feldern für Titel und Jahr. Die Art der Quelle wird in "\$kind" gespeichert. Dabei werden nur die Namen verwendet, die in der XML-Datei zur Kennzeichnung der Quellenart benutzt werden. Die Sortierung wird durch die Variablen \$sortierung und \$reihe definiert. Der Anzeigestil wird in der Variablen "\$style" gespeichert. Die Formularfelder enthalten je nach Art vorgegebene Werte, die im Value-Teil der Formularfeld -Deklaration auftauchen. Das ist für die Felder Quellenart, Sortierung, Sortierungsreihenfolge und Anzeigestil der Fall. Für Textfelder, wie das für den Autornamen, gelten keine derartigen Inhaltsbeschränkungen.

"\$bib" enthält die Angabe der Bibliographiedatei. Da in dieser Diplomarbeit nur eine Beispieldatei angezeigt werden soll, genügt es, diese Angabe für den Anwender unsichtbar in das Skript zu integrieren. Falls einmal die Notwendigkeit besteht weitere Bibliographien anzuzeigen, kann man das versteckte Feld sichtbar machen und weitere Auswahlfelder hinzufügen.

```
<input type="hidden" name="bib" value="xml">
```

Die Formularfelder für Auturname, Titel und Jahr sind Textfelder und haben keine vorgegebenen Werte für "\$lastname", "\$title", "\$year":

```
<table border=0 width="50%">
  <tr><td>Auturname (Nachname)</td><td align="left"><input
type="text" size="50" name="lastname"></td></tr>
  <tr><td>Titel</td><td align="left"><input type="text" size="50"
name="title"></td></tr>
  <tr><td>Jahr</td><td align="left"><input type="text" size="50"
name="year"></td></tr>
</table>
```

Das Formularfeld "Quellenart" mit den vorgegebenen Inhalten der Variablen "\$kind":

```
<select name="kind">
  <option selected="selected" value="">alle</option>
  <option value="book">Buch</option>
  <option value="article">Zeitschriftenartikel</option>
  <option value="inbook">Teil eines Buches</option>
  <option value="inproceedings">Teil eines Tagungsberichtes</option>
  <option value="incollection">Teil einer Aufsatzsammlung</option>
  <option value="techreport">Techreport</option>
  <option value="mastersthesis">Diplomarbeit</option>
```

```
<option value="unpublished">unver&#246;ffentlicht</option>
</select>
```

Das Formularfeld Sortierung mit den vorgegebenen Inhalten der Variablen "\$sortierung":

```
<select name="sortierung">
<option selected="selected" value="Autorname">Autorname</option>
<option value="Jahr">Erscheinungsjahr</option>
</select>
```

Das Formularfeld Sortierreihenfolge mit den vorgegebenen Inhalten der Variablen "\$reihe":

```
<select name="reihe">
<option selected="selected" value="aufsteigend">aufsteigend</option>
<option value="absteigend">absteigend</option>
</select>
```

Das Formularfeld "Anzeigestil" mit den vorgegebenen Inhalten der Variablen "\$style":

```
<input type="radio" name="style" value="din" checked>DIN 1505<br>
<input type="radio" name="style" value="apa">APA<br>
<input type="radio" name="style" value="harvard">Harvard<br>
```

Ein verstecktes Feld dient zur Überprüfung, ob das Formular vom Anwender neu aufgerufen oder bereits abgeschickt wurde. Je nach dem Wert der Variablen "\$sent" zeigt der Browser das Suchformular oder die Ergebnisliste an. Der Code für dieses Feld lautet:

```
<input type="hidden" name="sent" value=1>
```

Das PHP-Formular enthält neben den Formularfeldern ein Skript zur Überprüfung der Variablen, die an den XSL-Prozessor übergeben werden. Das Skript legt fest, dass über der Anzeige der Ergebnisliste noch einmal die eingegebenen Begriffe angezeigt werden sollen. Dies erfolgt über den Echo-Befehl.

```
if ($sent and $style) {
echo "<h3>Sie haben folgende Auswahl getroffen:</h3><p>";
echo "<b>Autorname:</b> $lastname<br>";
```

```
echo "<b>Titel:</b> $title<br>";
echo "<b>Erscheinungsjahr:</b> $year<br>";
echo "<b>Literaturart:</b> $kind<br>";
echo "<b>Sortierung:</b> $sortierung<br>";
echo "<b>Reihenfolge:</b> $reihe<br>";
echo "<b>Anzeigestil:</b> $style<br></p>";
```

Je nach dem Inhalt der Variable `$style` wird einer neuen Variablen, "`$xslname`" ein anderer Wert zugewiesen. Dieser Wert enthält den Dateinamen des XSL-Stylesheets. Die Variable "`$xslname`" dient dazu, dem Prozessor mitzuteilen, welche XSL-Datei verarbeitet werden soll.

Falls einmal der Fall besteht, dass eine weitere Bibliographie-Datei angezeigt werden soll, existiert die Variable `$bibname`. Diese weist gemäß dem Inhalt der im Moment versteckten Bibliographie-Dateiauswahl einen anderen Dateinamen für die neue Bibliographie zu.

Um die Anzeige der ausgewählten Werte im Klartext anzuzeigen, benötigt man `$sortierung` und `$reihe`. Um später die benötigten Werte an den XSL-Prozessor weiterzugeben, werden den Variablen `$sort` und `$order` je nach dem Inhalt von `$sortierung` und `$reihe` andere Werte zugewiesen. Ähnlich funktioniert die Anzeige Werte für die Variablen `$bib` und `$style`, die je nach Inhalt den Variablen "`$xslname`" und "`$bibname`" andere Werte zuweisen.

So kann man für "`$xslname`" und "`$bibname`" die Namen der Dateien und ihrer Pfade auch an einer einzigen Stelle im Skript bearbeiten.

```
if ($bib == "xml") {$bibname = "bibliography.xml";}
if ($style == "DIN") {$xslname = "din.xsl";}
if ($style == "APA") {$xslname = "apa.xsl";}
if ($style == "Harvard") {$xslname = "harvard.xsl";}
if ($sortierung == "Autornamen") {$sort = "lastn";}
if ($sortierung == "Jahr") {$sort = "year";}
if ($reihe == "aufsteigend") {$order = "ascending";}
if ($reihe == "absteigend") {$order = "descending";}

```

Das Skript überprüft auch, ob kein einziges Suchfeld ausgefüllt wurde. Ist das der Fall wird die Variable "`$fields`" auf 1 gesetzt. Sie bewirkt, dass, obwohl nichts eingegeben wurde, die gesamte Bibliographie ausgegeben wird. Durch die Verwendung von "Radiobutton"-Eingabefeldern für die Auswahl des Anzeigestils und der Sortierung wird für deren Variablen immer ein Wert weitergegeben, so dass die Ausgabe auf jeden Fall mit einem Stylesheet erfolgt. Der Anzeigestil bleibt also auch veränderbar, wenn die gesamte Bibliographie ausgegeben wird.

```
if ($lastname or $title or $year or $kind) {$fields = "0";}
else {$fields = "1";}
```

Der Code für das gesamte PHP-Formular ist im Anhang C aufgeführt.

7.4 Aufruf des XSLT-Prozessors

Das PHP-Skript ruft den XSLT-Prozessor Sablotron auf und übergibt an ihn die zur Verarbeitung benötigten Informationen. Diese Informationen werden von den Variablen des PHP-Skripts gespeichert. Der PHP-Befehl zum Aufruf des Prozessors lautet:

```
$xh = xslt_create();
```

Beim Aufruf des Prozessors werden Variablen, die zur Verarbeitung mit den XSL-Stylesheets benötigt werden, übergeben und die zu verarbeitende XSL- und XML-Datei angegeben.

Die zu übergebenden Variablen müssen in einem Array gespeichert werden, der als Parameter an den Prozessor übergeben wird. Dieser Array heißt "\$param". Innerhalb des Arrays werden Variablen, die im XSL-Stylesheet verwendet werden sollen, die Werte aus den Formularfeldern zugewiesen.

Der Array "\$param" lautet:

```
$param = array ('kind' => $kind, 'lastname' => $lastname, 'title' =>
$title, 'year' => $year, 'test' => $fields, 'sort' => $sort, 'order'
=> $order );
```

Die zu verwendende XSL-Datei muss dem Prozessor durch Pfad und Dateinamen angegeben werden. Wenn die XSL-Datei im gleichen Verzeichnis liegt wie die PHP-Datei mit dem Suchformular, genügt jedoch die Angabe des Dateinamens. Die Angabe der XML-Datei erfolgt nach dem gleichen Prinzip.

Die Verarbeitung wird durch den Befehl "xslt_process" gestartet. Der Prozessor greift auf XML-Datei, XSL-Datei und den Array für die Parameter zu.

```
$result = xslt_process($xh, $bibname, $xslname, NULL, array(),
$param);
```

\$xh ist ein Handle für den erzeugten Prozessor. Es folgen die Variablen für die Dateinamen der XML-Datei und der XSL-Datei. Ein NULL-Wert und ein leerer Array stehen für nicht benötigte Eingabewerte, die beim Starten des Prozesses trotzdem vorhanden sein müssen. Abschließend wird die Variable "\$param" angegeben, die auf den Parameter-Array verweist.

Die Variable "\$result" enthält nach einer erfolgreichen Verarbeitung einen String, der als HTML-Code dargestellt werden kann.

Der gesamte Aufruf des Prozessors und die Variablenübergabe lautet im PHP-Skript:

```
$xh = xslt_create();  
$param = array ('kind' => $kind, 'lastname' => $lastname, 'title' =>  
$title, 'year' => $year, 'test' => $fields, 'sort' =>$sort, 'order'  
=>$order );  
$result = xslt_process($xh, $bibname, $xslname, NULL, array(),  
$param);
```

7.5 Verwendung der XSL-Stylesheets

In den XSL-Stylesheets liegt die Grundlage der Sortier- und Suchfunktion. Mit den Anweisungen aus dem Stylesheet kann der Prozessor einen Ergebnisbaum aus den Daten der XML-Datei zusammenstellen. In Kapitel 7.5.2 wird die Durchführung der Sortierfunktion dargestellt. In den folgenden Kapiteln wird gezeigt, wie die einzelnen Programmteile für die Anzeige von Autor und den übrigen Teilen einer Quellenangabe aufgebaut sind und zusammenspielen. Im Kapitel 7.5.5 wird die Umsetzung der Suchfunktion erklärt.

7.5.1 Grundlagen

Um das Verständnis der Codebeispiele zu erleichtern, sollen hier grundlegende Informationen zu BibTeXML-Dateien und XSL gegeben werden.

Knoten und Elemente

Eine BibTeXML-Datei enthält einen Root-Knoten. In der verwendeten Beispieldatei ist es der Knoten "<bibliography>". Gleichzeitig enthält dieser Knoten ein Element, das "bibliography" heißt. Das Element wird so dargestellt: Es beginnt mit einem öffnenden Tag und endet mit einem schließenden (<bibliography> </bibliography>). Zwischen diesen Element-Tags dürfen beliebig viele Bibliographieeinträge stehen. Alle Elemente, aus denen ein Bibliographieeintrag besteht, sind mit den Element-Tags <bibitem> und </bibitem> eingeschlossen. Innerhalb der <bibitem>-Tags können verschiedene andere Elemente, wie Autor, Titel oder Jahr stehen.

Beispiel für eine Bibliographie mit nur einem Eintrag:

```
<bibliography>  
  <bibitem type="book" label="Alexander:1977">  
    <author>  
      <firstname>C.</firstname>  
      <lastname>Alexander</lastname>  
    </author>  
    <title>A Pattern Language</title>
```

```

    <publisher>Oxford University Press</publisher>

    <year>1977</year>

  </bibitem>
</bibliography>

```

Kommt eine zweite Quellenangabe hinzu, wird sie in einem neuen `<bibitem>`-Element eingebettet. So ergibt sich aus den Elementen eine Baumstruktur mit Verzweigungen. Jedes Element stellt eine Verzweigung des Elternelements dar.

Elementattribute

Ein `<bibitem>`-Element besitzt neben untergeordneten Elementen auch Attributwerte. Die Attributwerte für `<bibitem>` heißen "type" und "label". "type" enthält einen vorgegebenen Wert für die Art der Quellenangabe. Die Vorgabewerte stammen aus der DTD, in der zusätzlich festgelegt wurde, dass es zwingend notwendig (`#REQUIRED`) ist, einen Wert für die Attribute "type" und "label" anzugeben. Erlaubt sind in der Attributliste für `<bibitem>` (`<!ATTLIST bibitem`) der DTD für das Attribut "type" folgende Werte:

```

<!ATTLIST bibitem
  type  ( article | book | booklet |
          manual | techreport |
          mastersthesis | phdthesis |
          inbook | incollection |
          proceedings | inproceedings |
          conference | unpublished | misc ) #REQUIRED

```

Das Zeichen " | " bedeutet, dass eine ODER-Auswahl vorliegt. Pro `<bibitem>`-Element darf nur einer der Werte dieser Liste eingetragen werden.

Das Attribut Label ermöglicht bei konsequenter Nutzung eine eindeutige Identifizierung jedes einzelnen Bibliographieeintrags. Im Beispiel setzt sich der Inhalt aus dem Textinhalt der Elemente `<lastname>` und `<year>` zusammen ("Alexander:1977"). Diese beiden Angaben genügen allerdings nicht, wenn die Bibliographiedatei sehr groß ist. Es müssten weitere Inhaltstexte hinzugefügt werden. Anbieten würden sich z. B. die ersten Worte des Titels.

Baumstruktur

Der Element-Tag, der für den Eintrag des Autornamens gedacht ist, kann weitere untergeordnete Elemente enthalten: `<firstname>`, `<middlename>` und `<lastname>`. Bei mehreren Autoren dürfen auch mehrere `<author>`-Element-Tags verwendet werden:

```

<bibliography>
  <bibitem type="book" label="Alexander:1977">
    <author>
      <firstname>C.</firstname>
      <lastname>Alexander</lastname>
    </author>
    <author>
      <firstname>S.</firstname>
      <lastname>Ishikawa</lastname>
    </author>
    <author>
      <firstname>M.</firstname>
      <lastname>Silverstein</lastname>
    </author>
    <title>A Pattern Language</title>
    <publisher>Oxford University Press</publisher>
    <year>1977</year>
  </bibitem>
</bibliography>

```

Eine so entstandene Baumstruktur kann zur besseren Übersicht einmal grafisch dargestellt werden. In der folgenden Grafik wird zusätzlich ein zweites <bibitem>-Element mit weiteren Verzweigungen angedeutet. Leere Felder (..) stehen für weitere mögliche Elemente:

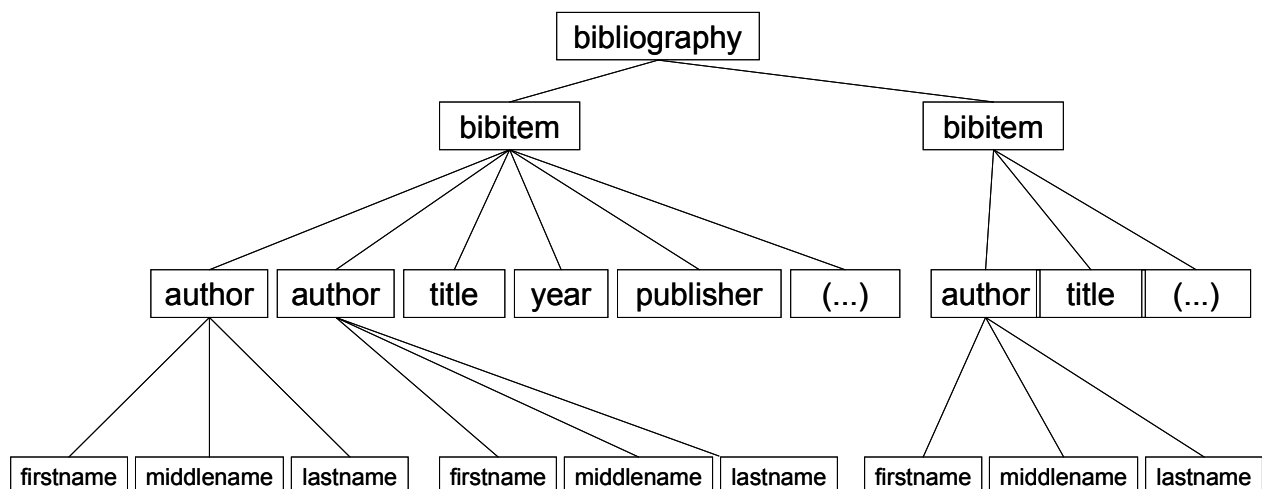


Abbildung 25: Beispiel einer Baumstruktur von BibTeX XML

Diese Struktur liefert die BibTeX-Grundlage der folgenden Codebeispiele. Eine XSL-Datei ist ähnlich aufgebaut wie eine XML-Datei. Unbedingt benötigt werden Tags, die geöffnet und wieder geschlossen werden. Das gilt auch für Transformations-Anweisungen. Eine Besonderheit bei XSL besteht darin, dass ein Tag in einem einzigen Schritt geöffnet und geschlossen werden kann. Die Zeichenfolge ">" am Ende des öffnenden Tags schließt ihn nach seiner Ausführung unmittelbar.

```
<xsl:sort select="/*/lastname"/>
```

Dieses Vorgehen ist jedoch nicht immer sinnvoll, da manche Teile des Codes von den Transformations-Anweisungen eingefasst werden sollten, wie z. B. für den Befehl `<xsl:for-each>`:

```
<xsl:for-each select="./bibitem">
  <!-- Code für alle "bibitem"-Elemente -->
</xsl:for-each>
```

7.5.2 Realisierung der Sortierfunktion

Um eine Sortierung der Anzeige zu erreichen, werden `<xsl:for-each>`- und `<xsl:sort>`-Befehle verwendet. Mit `<xsl:for-each>` kann jeder Elementtag `<bibitem>` mit allen untergeordneten Knoten erreicht werden. `<xsl:sort>` benötigt die Angabe, nach welchen Elementen sortiert werden soll. Hier sind es die Elemente `<lastname>`, `<firstname>` und `<year>`. Um eine Sortierung aller Bibliographieinträge nach Autor und anschließend nach Jahr zu erreichen, folgen die Befehle direkt aufeinander.

```
<xsl:for-each select="./bibitem">
  <xsl:sort order="descending" select="/*/lastname"/>
  <xsl:sort order="descending" select="/*/firstname"/>
  <xsl:sort order="descending" select="./year"/>
```

Um die Sortierreihenfolge zu berücksichtigen, werden mit `<xsl:if>` Tests durchgeführt. In `<xsl:sort>`-Befehlen dürfen keine Variablen verwendet werden. Daher muss für jede mögliche Sortierung und Sortierreihenfolge eine eigene Anweisung gegeben werden.

Hier ein Beispiel für die Anweisung zu einer absteigenden Sortierung nach Verfassername:

```
<!-- Sortieren nach Verfassernachname anschliessend nach Jahr -->
<xsl:if test="$sort = 'lastn'">
  <!-- absteigend -->
  <xsl:if test="$order = 'descending'">
    <xsl:for-each select="./bibitem">
      <xsl:sort order="descending" select="/*/lastname"/>
```

```
<xsl:sort order="descending" select="/*/firstname"/>
<xsl:sort order="descending" select="./year"/>
```

Dem Code folgen die Transformationsanweisungen der Elemente der Quellenangaben.

7.5.3 Anzeige eines oder mehrerer Autornamen

Die Anzeige von Autornamen ist von der Anzeige der restlichen Teile der Quellenangabe gesondert zu betrachten, weil innerhalb einer Quellenangabe mehrere Autoren vorkommen können. Die anderen Elemente wie Titel, Jahr oder Verlag kommen jedoch pro Quellenangabe nur einmal vor.

Alle Code-Beispiele beziehen sich auf die Anzeige nach dem Harvard-Stil, daher muss berücksichtigt werden, dass hier angegebene Formatierungszeichen wie Kommas oder Punkte für andere Stilarten nicht gelten.

Um einem Element mit XSL spezielle Anzeigeformen und Anweisungen zuzuweisen, benötigt man Templates. Um das Element `<author>` anzusprechen und zu formatieren, wird mit einem `<xsl:template>`-Befehl das Element `<author>` ausgewählt.

```
<xsl:template match="author">
```

Das bedeutet, innerhalb der Tags `<xsl:template>` und `</xsl:template>` findet man Anweisungen, die sich nur auf Elemente beziehen können, die gleich dem `<author>`-Element oder untergeordnet sind.

Templates können für jeden Element-Knoten der XML-Datei angegeben werden. In diesem Beispiel gibt es Templates für den Knoten `<bibliography>`, `<bibitem>` und `<author>`. In der gesamten XSL-Datei existieren weitere Templates, z. B. `<title>` oder `<address>`.

Die übergeordneten Templates beinhalten neben dem Aufruf der untergeordneten Templates auch Anweisungen für die HTML-Anzeige. So enthalten die Templates für den Root-Knoten und für `<bibliography>` auch die HTML-Tags `<table>`, `<tr>`, `<td>` und `<div>`, die in einem Browser eine HTML-Tabelle und zusätzlich einen Tag für die Formatierung mit CSS (Cascading Style Sheets) anzeigen.

```
<xsl:template match="/">
    <font face="Arial">
        <xsl:apply-templates select="/bibliography"/>
    </font>
</xsl:template>

<xsl:template match="/bibliography">
```

```

<h1>Bibliographie nach Harvard-Stil</h1>

<table>

<tr>

<td>

<div>

```

Nach Anwendung des Templates für <author> soll ein einzelner oder mehrere Autornamen nach der Form

Nachname, Vorname Zweitname

bzw. bei zwei Autornamen

Nachname, Vorname Zweitname and Nachname, Vorname Zweitname

angezeigt werden.

Um herauszufinden, ob es einen oder mehrere Autoren gibt, muss gezählt werden, wie viele <author>-Elemente im <bibitem>-Element vorhanden sind. Wenn es mehr als einer ist, muss eine andere Formatierung gewählt werden als für einen einzigen. Die Überprüfung erfolgt mit den Befehlen <xsl:choose> und <xsl:when>, welches mit einem Knotentest verbunden ist.

```

<xsl:template match="author">

  <xsl:choose>

    <xsl:when test="count(ancestor::bibitem/author)>1">

```

Trifft es zu, dass es mehr als ein <author>-Element gibt, werden die nachfolgenden Programmzeilen benutzt.

Zuerst wird der Nachname aufgerufen. Da sich das aktuelle Template auf den Knoten <author> bezieht, wird der Knoten <lastname> als Child-Knoten aufgerufen⁴.

```

<xsl:value-of select="child::lastname"/>

```

Anschließend wird mit

```

<xsl:if test="count(firstname)=1">

```

getestet, ob ein Vorname eingetragen ist. Ist das der Fall, wird ein Komma eingefügt und der Inhalt des Elements <firstname> angezeigt, gefolgt von einem Leerzeichen. </xsl:if> schließt die If-Bedingung.

⁴ Der Inhalt von Child-Knoten wie <lastname> kann auch mit dem Befehl ".//Knotenname" aufgerufen werden, z. B.: <xsl:value-of select=".//lastname"/>.

```

<xsl:text>, </xsl:text>

<xsl:value-of select="child::firstname"/>

<xsl:text> </xsl:text>

</xsl:if>

```

Falls ein zweiter Vorname angegeben ist, wird dieser angefügt.

```

<xsl:if test="count(middlename)=1">

<xsl:value-of select="child::middlename"/>

</xsl:if>

```

Die oben angegebenen Codeteile funktionieren bis auf den Test zur Anzahl von `<author>`-Knoten auch für die Anzeige eines einzigen Autoren. Um mehrere Autornamen zu verbinden, fehlt noch der folgende Teil:

Zwischen den Namen mehrerer Autoren soll das Verbindungswort "and" eingefügt, jedoch nach dem letzten Autor weggelassen werden. Dazu wird die Position des `<author>`-Elements innerhalb des `<bibitem>`-Knotens überprüft. Solange das aktuelle `<author>`-Element nicht das letzte der Liste ist wird ein "and" eingefügt.

```

<xsl:if test="position() != last()">

<xsl:text> and </xsl:text>

</xsl:if>

</xsl:when>

```

Um das Template zu vervollständigen, muss nach diesen Codeteilen ein weiterer Knotentest für einen einzigen Autor, der Code zur Anzeige der Namensteile und Schluss-tags für den Auswahlbefehl `<xsl:choose>` (`</xsl:choose>`) und das Template (`</xsl:template>`) eingefügt werden.

7.5.4 Anzeige der übrigen Teile einer Quellenangabe

Zu einer Quellenangabe gehören neben dem Autornamen noch Elemente wie Titel, Jahr, Verlag, Verlagsort, Ausgabe, Zeitschriftentitel, Bandtitel, Herausgeber oder Seitenzahl.

Zur richtigen Darstellung der Teile ist es wichtig herauszufinden, welche Art der Quellenangabe vorliegt. Je nach Art gelten nach den verschiedenen Anzeigestilen besondere Regeln für die Darstellung. Eine Monographie wird anders dargestellt als ein Zeitschriftenartikel, der zusätzlich zu Angaben zum Artikel auch Informationen zur Zeitschrift enthalten muss.

Um herauszufinden, um welche Art der Quelle es sich handelt, benötigt man einen Test, der je nach dem Wert des Attributs "type" des `<bibitem>`-Elements Teile eines speziellen Templates aufruft. Das Template enthält je nach Art der vorhandenen Elemente unterschiedliche Transformationsaufrufe.

Der Test für die Art der Quelle wurde auf folgende Weise realisiert: Wenn das Attribut "type" den Wert "article" enthält, soll das Template "@type" aufgerufen werden.

```
<!-- wenn Zeitschriftenartikel -->
<xsl:when test="@type='article'">
    <xsl:apply-templates select="@type"/>
</xsl:when>
```

In dem Template "@type" erfolgt ein ähnlicher Test, der eine Wahl zwischen Template-teilen für unterstützte Attributwerte erlaubt.

Für den Attributwert "article" gilt der folgende Teil des Templates:

```
<xsl:template match="@type">
    <xsl:choose>
        <!-- article -->
        <xsl:when
test="ancestor::bibitem[@type='article']">
```

Ist das Ergebnis dieses Tests wahr ("TRUE"), werden die nachfolgenden Codezeilen angewendet. Für jedes erwartete Element wird getestet, ob es vorhanden ist. Wenn ja, werden neben dem Inhalt auch Satzzeichen oder Formatierungselemente für die HTML-Ausgabe aufgerufen.

Dieses Schema läuft für alle Elemente ähnlich ab, deshalb genügt an dieser Stelle ein Beispiel für das Element Jahr:

```
<xsl:if test="count(ancestor::bibitem/year)=1">
<xsl:text> </xsl:text>
<xsl:value-of select="ancestor::bibitem/year"/>
<xsl:text>, </xsl:text>
</xsl:if>
```

Dieses Schema gilt auch für die folgenden Elemente bei "article":

- Title
- Journal
- Band, wenn vorhanden
- Nummer
- Monat
- Seiten

Für alle weiteren Arten von Quellen und deren "type"-Werte in <bibitem> folgt die Transformation und Formatierung einem parallelen Schema. Unterschiede ergeben

sich lediglich daraus, dass Inhalte des <bibitem>-Elements je nach Art der Quelle variieren können. Z. B. wird bei einer Quellenangabe zu einem Buch das Element Journal, das den Titel einer Zeitschrift enthält, nicht verwendet. <journal> muß also nicht in diesem Teil des Templates mit Transformations- und Formatierungsanweisungen versehen und kann weggelassen werden.

7.5.5 Umsetzung der Suchfunktion

Die Suchfunktion ermöglicht eine Einschränkung der Anzeige auf bestimmte Inhalte. Weil XSL aber solche Einschränkungen nicht ohne weiteres unterstützt, müssen für die Suche besondere Codeteile verwendet werden. Die Suche basiert auf den Werten der Formularfelder, die über das PHP-Skript im Array "\$param" an den XSL-Prozessor weitergegeben werden. Dieser verwendet XSL-Sheets, die Variablen enthalten. In die Variablen kann der Prozessor die übergebenen Werte einsetzen. Mit den unterschiedlichen Eingabewerten können die Transformationsanweisungen im XSL-Stylesheet und damit auch die Ausgabe des Prozessors variiert werden.

Die Variablen im XSL-Stylesheet müssen vor der Benutzung deklariert werden. Das erfolgt zu Beginn des Codes, um zu gewährleisten, dass sie global aus jedem Template ansprechbar sind.

```
<xsl:param name="lastname" select="''" />
<xsl:param name="title" select="''" />
<xsl:param name="year" select="''" />
<xsl:param name="kind" select="''" />
<xsl:param name="test" select="''" />
<xsl:param name="sort" select="''" />
<xsl:param name="order" select="''" />
```

Zur Suche wird getestet, ob vom PHP-Skript Variablen übergeben wurden. Das geschieht durch die Bestimmung der Stringlänge. Ist die Länge des übergebenen Strings größer 0 (">0"), soll der dieser Zeile nachfolgende Code benutzt werden.

```
<xsl:if test="string-length($title )>0">
```

Mit solchen Tests werden alle möglichen Kombinationen von Variablen abgehandelt. Eine Einschränkung der Anzeige auf alle Quellenangaben, deren Inhalt an einer Stelle mit dem der Variablen übereinstimmt, und so ein Suchergebnis darstellen, wird mit

```
<xsl:if test="title[contains(string(.), $title)]">
```

erreicht.

Hier wird getestet, ob Variablen für \$kind, \$title und \$year angegeben wurden.

```
<xsl:choose>
  <xsl:when test="string-length($kind )=0">
    <!-- Titel und Jahr vorhanden -->
    <xsl:if test="string-length($title )>0">
      <xsl:if test="title[contains(string(.), $title)]">
        <xsl:if test="string-length($year )>0">
          <xsl:if test="year[contains(string(.), $year)]">
```

Der den Tests nachfolgende Code wird in diesem Beispiel nur benutzt, wenn Titel und Jahr angegeben wurden, jedoch keine Auswahl für die Art der Quellenangabe getroffen wurde.

XSL unterstützt keine komplexen Abfragefunktionen, daher ist der Code für die Suche auf Tests beschränkt, die simple "wahr/falsch"-Überprüfungen vornehmen. Das geschieht durch die Befehle <xsl:choose>, <xsl:when> und <xsl:if>.

Eine schematische Übersicht über alle Tests zeigt die Komplexität der Suchfunktion:

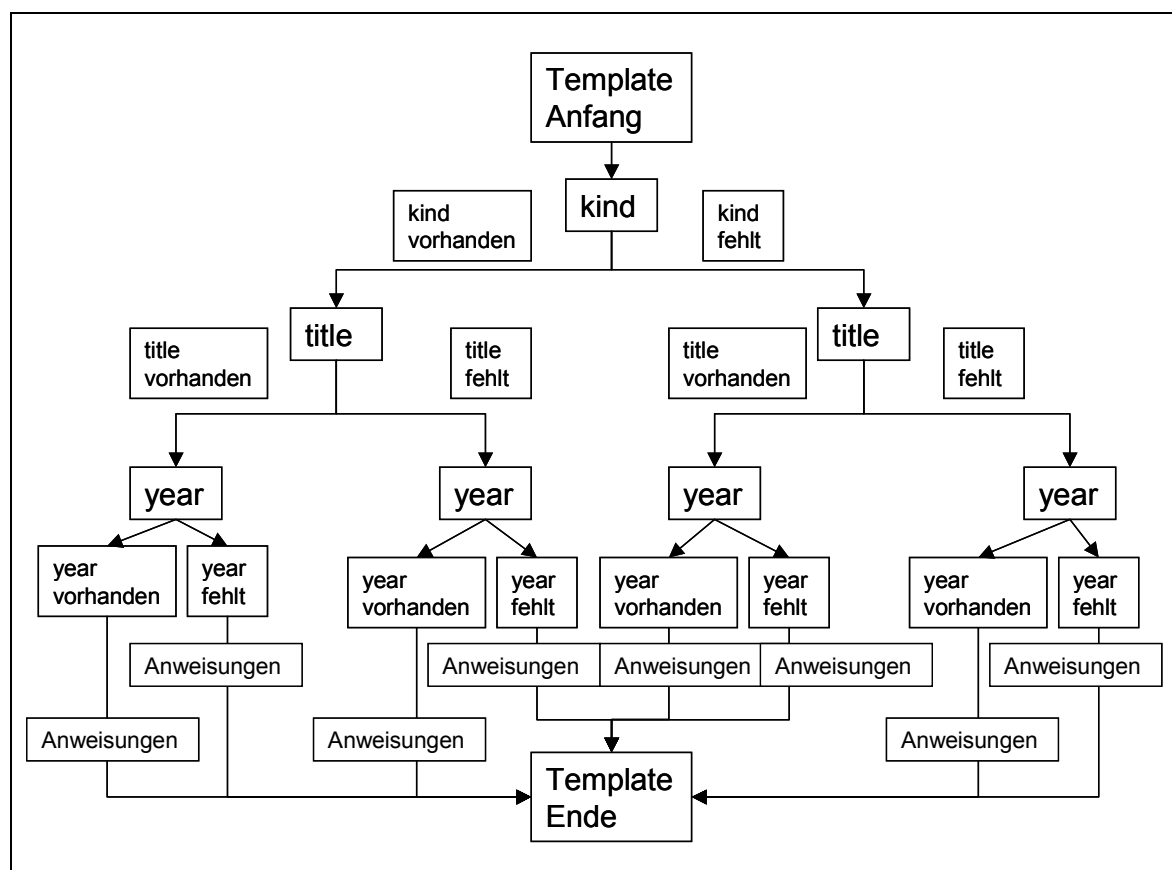


Abbildung 26: Darstellung der Struktur der Suchfunktion

Dabei wird für jede mögliche Variablenkombination zusätzlich getestet, ob die <author>-Elemente des <bibitem>-Elements den gesuchten Autornamen-String enthal-

ten und falls es eine Einschränkung zur Art der Quellenangabe gibt, nur diese anzuzeigen. Dieser Test lautet:

```
<xsl:if test="@type= $kind">
<xsl:if test="author[contains(string(/lastname), $lastname)]">
```

Je nach der Kombination der eingegebenen Suchvariablen wird eine andere Reihe von Transformationsanweisungen benutzt, die sich jedoch größtenteils ähneln. Als Beispiel hier die Anweisungen für die Variablenkombination Titel, Jahr, Nachname und allen Literaturarten:

```
<xsl:if test="string-length($title )>0">
<xsl:if test="title[contains(string(/), $title)]">
<xsl:if test="string-length($year )>0">
<xsl:if test="year[contains(string(/), $year)]">
<xsl:if test="author[contains(string(/lastname), $lastname)]">
    <table>
        <tr>
            <td>
                <div>

                    <xsl:apply-templates select="author"/>

                    <xsl:choose>

                        <!-- wenn selbstaendige Einheit -->
                        <xsl:when test="@type='book'">
                            <xsl:apply-templates select="year"/>
                            <xsl:text>, </xsl:text>

                            <xsl:if test="count(title)=1">
                                <xsl:apply-templates select="title"/>
                                <xsl:text>. </xsl:text>
                            </xsl:if>

                            <xsl:if test="count(address)=1">
                                <xsl:apply-templates select="address"/>
                                <xsl:text>: </xsl:text>
                            </xsl:if>

                            <xsl:if test="count(publisher)=1">
                                <xsl:apply-templates select="publisher"/>
                            </xsl:if>
                            <xsl:text>. </xsl:text>

                        </xsl:when>

                        <!-- wenn Zeitschriftenartikel -->
                        <xsl:when test="@type='article'">
```



```

        <xsl:apply-templates select="@type"/>
      </xsl:when>

      <!-- wenn inproceedings -->
      <xsl:when test="@type='inproceedings'">
        <xsl:apply-templates select="@type"/>
      </xsl:when>

      <!-- wenn in Buch -->
      <xsl:when test="@type='inbook'">
        <xsl:apply-templates select="@type"/>
      </xsl:when>

      <!-- wenn incollection -->
      <xsl:when test="@type='incollection'">
        <xsl:apply-templates select="@type"/>
      </xsl:when>

      <!-- wenn unpublished -->
      <xsl:when test="@type='unpublished'">
        <xsl:apply-templates select="@type"/>
      </xsl:when>

      <!-- wenn Techreport -->
      <xsl:when test="@type='techreport'">
        <xsl:apply-templates select="@type"/>
      </xsl:when>

      <xsl:when test="@type='mastersthesis'">
        <xsl:apply-templates select="@type"/>
      </xsl:when>

    </xsl:choose>

  </div>
</td>
</tr>
</table>
</xsl:if>

</xsl:if>
</xsl:if>

```

Im Anhang B ist der gesamte Code der Suchfunktion angegeben.

Um alle Einträge in der Bibliographiedatei anzuzeigen, benötigt man die Variable \$test. Sie wird auf den Wert 1 gesetzt, wenn bei der Überprüfung durch das PHP-Skript festgestellt wurde, dass statt der Suchfelder die Option "alle anzeigen" aktiviert wurde. Wenn die Variable einen String enthält, der länger als 0 ist, wird statt des Such-Templates "bibitem" ein Template verwendet, das alle Einträge anzeigt: "bibliography".

7.6 Ausgabe der Ergebnisliste

Nachdem der Prozessor die Eingabedaten verarbeitet hat, gibt er einen String aus, der im PHP-Formular angezeigt werden soll. Das erfolgt durch den Einsatz einer Variablen, \$result, in der der ausgegebene String gespeichert wird. Das PHP-Skript zeigt den Inhalt der Variablen nach dem Befehl "Print" und dem Aufruf der Variablen an.

```
if ($result) {
    print "<hr noshade>";
    print "<div>\n";
    print $result;
    print "</div>\n";
    print "<hr noshade>";
}
```

Falls bei der Verarbeitung im XSL-Prozessor ein Fehler passiert sein sollte, gibt das Skript eine Meldung zusammen mit der Fehlernummer des Prozessors aus.

```
else {
    print "Sorry, sample.xml could not be transformed by sample.xsl
into";
    print " the \$result variable the reason is that " .
xslt_error($xh) . " and the ";
    print "error code is " . xslt_errno($xh);
}
```

Nach der Verarbeitung wird der Prozessor freigegeben.

```
xslt_free($xh);
}
```

Wenn die Transformation der XML-Datei erfolgreich war, wird eine Tabelle angezeigt, die die Ausgabedaten enthält. Um diese Tabelle zu formatieren, wird ein CSS Style-sheet benötigt. In der CSS Datei wird festgelegt, dass die Einträge mit einem hängenden Einzug angezeigt werden. Diese Formatierung gilt für alle HTML-Tags, die "<div>" lauten. Dieser Tag wurde in den XSL-Stylesheets zusammen mit den HTML-Tags der Tabelle für die Anzeige eingetragen.

```
DIV { margin-left: 1cm; overflow:visible; text-indent: -1cm }
```

Das Endergebnis ist die Anzeige der Ergebnisliste. Sie enthält eine Auflistung der eingegebenen Werte und zeigt die Quellenangaben im ausgewählten Anzeigestil.

8 Zusammenfassung und Ausblick

Die Bibliographieanzeige ist über ein PHP-Formular im Internet aufrufbar. Eine Suchfunktion ermöglicht eine vielfältige Einschränkung in der Bibliographieanzeige. Sortierung und die Auswahl des Anzeigestils wurden umgesetzt. Die Bedienung des übersichtlichen Suchformulars durch den Anwender erfolgt leicht und problemlos.

So wird die Anzeige einer Quellenangabe schnell erreichbar und bereits im gewünschten Zitierformat nach DIN-, APA- oder Harvard-Stil ausgegeben. Das ermöglicht dem Anwender eine einfache Weiterverwendung der Quellenangabe.

Die Anzeige kann mit beliebigen Vorgaben in andere Homepages eingefügt werden.

Das hier realisierte Anzeigetool für Bibliographien, die im BibTeX-Format vorliegen, wurde für den Einsatz im Hochschulbereich entwickelt. Jedoch ist die Verwendung nicht auf diesen Bereich beschränkt. Dieses Tool ist für jede Bibliographie im BibTeX-Format geeignet. So ist z. B. eine Nutzung in Firmen- oder Behördenintranets genauso denkbar wie im privaten Bereich.

Eine Weiterentwicklung bietet sich für den Ausbau der Suchfunktion an. Unter Verwendung der XML-Abfragesprache Xquery könnten weitere spezialisierte Funktionen, wie z. B. die Verwendung von Wildcards oder Bool'schen Ausdrücken, eingerichtet werden.

Anhang A

Beispielcode einer BibTeXML-Datei

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE file PUBLIC "-//MCQM//DTD BibTeXML V0.1//EN" "bibteXML.dtd">

<bibliography>

<bibitem type="book" label="Alexander:1964">
<author>
<firstname>C.</firstname>
<lastname>Alexander</lastname>
</author>
<title>Notes on the Synthesis of Form</title>
<publisher>Harvard University Press</publisher>
<year>1964</year>
</bibitem>

<bibitem type="article" label="Alexander:1971">
<author>
<firstname>C.</firstname>
<lastname>Alexander</lastname>
</author>
<title>A Refutation of Design Methodology (Interview with Max
Jacobson)</title>
<journal>Architectural Design</journal>
<year>1971</year>
<month>December</month>
</bibitem>

<bibitem type="book" label="Alexander+Silverstein+Angel:1975">
<author>
<firstname>C.</firstname>
<lastname>Alexander</lastname>
</author>
<author>
<firstname>M.</firstname>
<lastname>Silverstein</lastname>
</author>
<author>
<firstname>S.</firstname>
<lastname>Angel</lastname>
</author>
<title>The Oregon Experiment</title>
<publisher>Oxford Universiy Press</publisher>
<year>1975</year>
</bibitem>
```

```
<bibitem type="book" label="Alexander+ Ishikawa+ Silverstein:1977">
<author>
  <firstname>C.</firstname>
  <lastname>Alexander</lastname>
</author>
<author>
  <firstname>S.</firstname>
  <lastname>Ishikawa</lastname>
</author>
<author>
  <firstname>M.</firstname>
  <lastname>Silverstein</lastname>
</author>
<title>A Pattern Language</title>
<publisher>Oxford University Press</publisher>
<year>1977</year>
</bibitem>
```

```
<bibitem type="book" label="Alexander:1979">
<author>
  <firstname>C.</firstname>
  <lastname>Alexander</lastname>
</author>
<title>A Timeless Way of Building</title>
<publisher>Oxford University Press</publisher>
<year>1979</year>
</bibitem>
```

```
<bibitem type="book" label="Alexander:1981">
<author>
  <firstname>C.</firstname>
  <lastname>Alexander</lastname>
</author>
<title>The Linz Cafe</title>
<publisher>Oxford University Press</publisher>
<year>1981</year>
</bibitem>
```

```
<bibitem type="book" label="Alexander:1985">
<author>
  <firstname>C.</firstname>
  <lastname>Alexander</lastname>
</author>
<title>The Production of Houses</title>
<publisher>Oxford University Press</publisher>
<year>1985</year>
</bibitem>
```

```
<bibitem type="book" label="Alexander:1987">
<author>
  <firstname>C.</firstname>
  <lastname>Alexander</lastname>
</author>
<title>A New Theory of Urban Design</title>
```

```

<publisher>Oxford University Press</publisher>
<year>1987</year>
</bibitem>

<bibitem type="article" label="Alexander:1991">
<author>
  <firstname>C.</firstname>
  <lastname>Alexander</lastname>
</author>
<title>Perspectives: Manifesto 1991</title>
<journal>Progressive Architecture</journal>
<year>1991</year>
<month>July</month>
</bibitem>

<bibitem type="article" label="Dovey:1990">
<author>
  <firstname>K.</firstname>
  <lastname>Dovey</lastname>
</author>
<title>The Pattern Language and its Enemies</title>
<journal>Design Studies</journal>
<year>1990</year>
<volume>11</volume>
<pages>3-9</pages>
</bibitem>

<bibitem type="article" label="Galle:1991">
<author>
  <firstname>P.</firstname>
  <lastname>Galle</lastname>
</author>
<title>Alexander Patterns for Design Computing: Atoms of Conceptual
Structure?</title>
<journal>Environment and Planning B: Planning and Design</journal>
<year>1991</year>
<volume>18</volume>
<pages>327-346</pages>
</bibitem>
</bibliography>

```

Beispielcode einer DTD für BibTeXML

```

<!--
- File: $Id: bibteXML.dtd,v 1.10 2002/06/20 12:54:39 zegerh Exp $
-
- This DTD expresses XML markup similar to the BibTeX language
- specified for LaTeX, or actually its content model.
-
- For documentation on BibTeX, see
-   http://www.ctan.org/tex-archive/biblio/bibtex/distrib/doc/
-
- Suggested invocation:

```

```

- <!DOCTYPE file PUBLIC "-//Oren Patashnik//DTD bibtex//EN"
-     "http://bibtexml.sf.net/bibtexml.dtd" >
- <file xmlns:bibtex="http://bibtexml.sf.net/">
-     ...
- </file>
-
- When used as a module within another DTD:
- <!ENTITY % bibtex PUBLIC
-     "-//Oren Patashnik//DTD bibtex//EN" "bibtexml.dtd">
- %bibtex;
- <!ELEMENT bibliography (title?, entry*) >
-
- Available from http://bibtexml.sf.net/
-
- This code originally developed by
- Vidar Bronken Gundersen, Zeger W. Hendrikse
- http://bibtexml.sf.net/
- Reuse and repurposing is approved as long as this
- notification appears with the code.
-
-->

<!-- ..... -->
<!-- Main structure -->

<!-- bibitem may contain one of the bibliographic types. -->
<!ELEMENT bibitem ( author*, title, booktitle, editors, journal, chapter) >

<!ATTLIST bibitem
    type ( article | book | booklet |
            manual | techreport |
            mastersthesis | phdthesis |
            inbook | incollection |
            proceedings | inproceedings |
            conference | unpublished | misc ) #REQUIRED
    label CDATA #REQUIRED >

<!-- bibliography is the document top element -->
<!ELEMENT bibliography ( bibitem )* >

<!-- bibitem untergeordnete Elemente -->
<!ELEMENT author (lastname, middlename, firstname)>

<!ELEMENT lastname (#PCDATA) >
<!ELEMENT middlename (#PCDATA) >
<!ELEMENT firstname (#PCDATA) >

<!ELEMENT title (#PCDATA)>
<!ELEMENT editors (#PCDATA) >
<!ELEMENT keywords (keyword+) >
<!ELEMENT keyword (#PCDATA) >
<!ELEMENT address (#PCDATA) >
<!ELEMENT booktitle (#PCDATA) >
<!ELEMENT chapter (#PCDATA) >

```

```

<!ELEMENT  edition      (#PCDATA) >
<!ELEMENT  editor       (#PCDATA) >
<!ELEMENT  howpublished  (#PCDATA) >
<!ELEMENT  institution   (#PCDATA) >
<!ELEMENT  journal      (#PCDATA) >
<!ELEMENT  month        (#PCDATA) >
<!ELEMENT  note         (#PCDATA) >
<!ELEMENT  number       (#PCDATA) >
<!ELEMENT  organization  (#PCDATA) >
<!ELEMENT  pages        (#PCDATA) >
<!ELEMENT  publisher    (#PCDATA) >
<!ELEMENT  school       (#PCDATA) >
<!ELEMENT  series       (#PCDATA) >
<!ELEMENT  title        (#PCDATA) >
<!ELEMENT  type         (#PCDATA) >
<!ELEMENT  volume       (#PCDATA) >
<!ELEMENT  year         (#PCDATA) >

<!-- These were not listed in the documentation for entry content, but
      - appeared in the list of fields in the BibTeX documentation -->

<!ELEMENT  annotate      (#PCDATA) >
<!ELEMENT  crossref     (#PCDATA) >
<!ELEMENT  key          (#PCDATA) >

<!-- ..... -->
<!-- Other popular fields
      -
      - From: http://www.ecst.csuchico.edu/~jacobsd/bib/formats/bibtex.html
      - BibTeX is extremely popular, and many people have used it to store
      - information. Here is a list of some of the more common fields:
      -
      - [affiliation] The authors affiliation.
      - [abstract] An abstract of the work.
      - [contents] A Table of Contents
      - [copyright] Copyright information.
      - [ISBN] The International Standard Book Number.
      - [ISSN] The International Standard Serial Number.
      - Used to identify a journal.
      - [keywords] Key words used for searching or possibly for annotation.
      - [language] The language the document is in.
      - [location] A location associated with the entry,
      - such as the city in which a conference took place.
      - [LCCN] The Library of Congress Call Number.
      - I've also seen this as lib-congress.
      - [mrnumber] The Mathematical Reviews number.
      - [price] The price of the document.
      - [size] The physical dimensions of a work.
      - [URL] The WWW Universal Resource Locator that points to the item being
      - referenced. This often is used for technical reports to point to the
      - ftp site where the postscript source of the report is located.
      -
      - When using BibTeX with LaTeX you need

```



```

- BibTeX style files to print these data.
-->

<!ELEMENT abstract      (#PCDATA) >
<!ELEMENT affiliation    (#PCDATA) >
<!ELEMENT contents      (#PCDATA) >
<!ELEMENT copyright     (#PCDATA) >
<!ELEMENT isbn           (#PCDATA) >
<!ELEMENT issn          (#PCDATA) >
<!ELEMENT language      (#PCDATA) >
<!ELEMENT lccn          (#PCDATA) >
<!ELEMENT location      (#PCDATA) >
<!ELEMENT mrnumber      (#PCDATA) >
<!ELEMENT price         (#PCDATA) >
<!ELEMENT size          (#PCDATA) >
<!ELEMENT url           (#PCDATA) >

<!-- Added by Zeger W. Hendrikse
- [category] Category of this bibitem
-->
<!ELEMENT category      (#PCDATA) >

<!-- ..... -->
<!-- Predefined/reserved character entities -->

<!ENTITY amp      "&#38;#38;">
<!ENTITY lt       "&#38;#60;">
<!ENTITY gt       "&#62;">
<!ENTITY apos     "&#39;">
<!ENTITY quot     "&#34;">

<!-- Added by Nora Kersten
- German Umlaute
-->

<!ENTITY Auml     "&#196;">
<!ENTITY Ouml     "&#214;">
<!ENTITY Uuml     "&#220;">
<!ENTITY auml     "&#228;">
<!ENTITY ouml     "&#246;">
<!ENTITY uuml     "&#252;">
<!ENTITY szlig    "&#223;">

<!-- ..... -->
<!-- End of bibtex dtd -->

```

Anhang B

Beispielcode der XSL-Datei für den Harvard-Anzeigestil

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE file PUBLIC "-//MCQM//DTD BibTeXML V0.1//EN" "bibteXML.dtd">
<!--
    Harvard formatting for BibTeXML version 1.0
    by Nora Kersten 2002
-->

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:param name="lastname" select="'" />
    <xsl:param name="title" select="'" />
    <xsl:param name="year" select="'" />
    <xsl:param name="kind" select="'" />
    <xsl:param name="test" select="'" />
    <xsl:param name="sort" select="'" />
    <xsl:param name="order" select="'" />

    <xsl:output method="html" encoding="UTF-8"
        doctype-public="-//W3C//DTD HTML 4.01 Transitional//EN"
        doctype-system="http://mars.iuk.hdm-
stuttgart.de/projekte/kersten/bibteXML.dtd"/>

    <xsl:template match="/">

        <font face="Arial">
            <xsl:apply-templates select="/bibliography"/>
        </font>

    </xsl:template>

    <xsl:template match="/bibliography">

        <h2>Bibliographie nach Harvard-Stil</h2>

        <!-- falls keine Suchbegriffe eingetragen -->
        <xsl:if test="string-length($lastname )=0">
        <xsl:if test="string-length($title )=0">
        <xsl:if test="string-length($year )=0">
        <xsl:if test="string-length($kind )=0">
        <xsl:if test="string-length($test )=1">
        <table>
```

```

<!-- Sortieren nach Verfassernachname anschliessend nach Jahr -->
    <xsl:if test="$sort = 'lastn'">
        <!-- absteigend -->
        <xsl:if test="$order = 'descending'">

            <xsl:for-each select="./bibitem">
<xsl:sort order="descending" select="/*/lastname"/>
<xsl:sort order="descending" select="/*/firstname"/>
<xsl:sort order="descending" select="./year"/>

                <tr>
                <td>

                    <div>

                        <!-- Bibliographische Angaben -->

                            <xsl:apply-templates select="author"/>

                            <xsl:choose>

                                <!-- wenn selbstaendige Einheit -->
                                <xsl:when test="@type='book'">
                                    <xsl:apply-templates select="year"/>
                                    <xsl:text>, </xsl:text>

                                    <xsl:apply-templates select="title"/>
                                    <xsl:if test="count(title)=1">
                                        <xsl:text>. </xsl:text>
                                    </xsl:if>

                                    <xsl:apply-templates select="address"/>
                                    <xsl:if test="count(address)=1">
                                        <xsl:text>: </xsl:text>
                                    </xsl:if>

                                    <xsl:apply-templates select="publisher"/>
                                    <xsl:if test="count(publisher)=1">
                                        <xsl:text>. </xsl:text>

                                    </xsl:if>

                                <xsl:text>. </xsl:text>

                                </xsl:when>

                                <!-- wenn Zeitschriftenartikel -->
                                <xsl:when test="@type='article'">
                                    <xsl:apply-templates select="@type"/>
                                </xsl:when>

                                <!-- wenn inproceedings -->
                                <xsl:when test="@type='inproceedings'">
                                    <xsl:apply-templates select="@type"/>
                                </xsl:when>

```

```

        <!-- wenn in Buch -->
        <xsl:when test="@type='inbook'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn incollection -->
        <xsl:when test="@type='incollection'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn unpublished -->
        <xsl:when test="@type='unpublished'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn Techreport -->
        <xsl:when test="@type='techreport'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <xsl:when test="@type='mastersthesis'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>
</div>
</td>
</tr>

</xsl:for-each>
</xsl:if>

    <xsl:if test="$order = 'ascending'">
<xsl:for-each select="./bibitem">
    <xsl:sort order="ascending" select="./*/lastname"/>
    <xsl:sort order="ascending" select="./*/firstname"/>
    <xsl:sort order="ascending" select="./year"/>

<tr>
<td>

    <div>

        <!-- Bibliographische Angaben -->
        <xsl:apply-templates select="author"/>

        <xsl:choose>

            <!-- wenn selbstaendige Einheit -->
            <xsl:when test="@type='book'">
                <xsl:apply-templates select="year"/>
                <xsl:text>, </xsl:text>

```

```
<xsl:apply-templates select="title"/>
<xsl:if test="count(title)=1">
  <xsl:text>. </xsl:text>
</xsl:if>

<xsl:apply-templates select="address"/>
<xsl:if test="count(address)=1">
<xsl:text>: </xsl:text>
</xsl:if>

<xsl:apply-templates select="publisher"/>
<xsl:if test="count(publisher)=1">

</xsl:if>

<xsl:text>. </xsl:text>

</xsl:when>

  <!-- wenn Zeitschriftenartikel -->
  <xsl:when test="@type='article'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn inproceedings -->
  <xsl:when test="@type='inproceedings'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn in Buch -->
  <xsl:when test="@type='inbook'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn incollection -->
  <xsl:when test="@type='incollection'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn unpublished -->
  <xsl:when test="@type='unpublished'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn Techreport -->
  <xsl:when test="@type='techreport'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <xsl:when test="@type='mastersthesis'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>
```

```

        </xsl:choose>
      </div>
    </td>
  </tr>
</xsl:for-each>
</xsl:if>
</xsl:if>

<!-- Sortieren nach Jahr anschliessend nach Autor -->

    <xsl:if test="$sort = 'year'">
      <xsl:if test="$order = 'descending'">

        <xsl:for-each select="./bibitem">
          <xsl:sort order="descending" select="./year"/>
          <xsl:sort order="descending" select="./*/lastname"/>
          <xsl:sort order="descending" select="./*/firstname"/>

        <tr>
        <td>

          <div>

            <!-- Bibliographische Angaben -->

            <xsl:apply-templates select="author"/>

            <xsl:choose>

              <!-- wenn selbstaendige Einheit -->
              <xsl:when test="@type='book'">
                <xsl:apply-templates select="year"/>
                <xsl:text>, </xsl:text>

                <xsl:apply-templates select="title"/>
                <xsl:if test="count(title)=1">
                  <xsl:text> . </xsl:text>
                </xsl:if>

                <xsl:apply-templates select="address"/>
                <xsl:if test="count(address)=1">
                  <xsl:text>: </xsl:text>
                </xsl:if>

                <xsl:apply-templates select="publisher"/>
                <xsl:if test="count(publisher)=1">
                  <xsl:text>
                </xsl:if>

                <xsl:text> . </xsl:text>

              </xsl:when>

```

```

        <!-- wenn Zeitschriftenartikel -->
        <xsl:when test="@type='article'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn inproceedings -->
        <xsl:when test="@type='inproceedings'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn in Buch -->
        <xsl:when test="@type='inbook'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn incollection -->
        <xsl:when test="@type='incollection'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn unpublished -->
        <xsl:when test="@type='unpublished'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn Techreport -->
        <xsl:when test="@type='techreport'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <xsl:when test="@type='mastersthesis'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>
</div>
</td>
</tr>

</xsl:for-each>
</xsl:if>

<xsl:if test="$order = 'ascending'">

<xsl:for-each select="./bibitem">
    <xsl:sort order="ascending" select="./year"/>
    <xsl:sort order="ascending" select="./*/lastname"/>
    <xsl:sort order="ascending" select="./*/firstname"/>

<tr>
<td>

        <div>

```

```
<!-- Bibliographische Angaben -->
  <xsl:apply-templates select="author"/>

  <xsl:choose>

    <!-- wenn selbstaendige Einheit -->
    <xsl:when test="@type='book'">
      <xsl:apply-templates select="year"/>
      <xsl:text>, </xsl:text>

      <xsl:apply-templates select="title"/>
      <xsl:if test="count(title)=1">
        <xsl:text>. </xsl:text>
      </xsl:if>

      <xsl:apply-templates select="address"/>
      <xsl:if test="count(address)=1">
        <xsl:text>: </xsl:text>
      </xsl:if>

      <xsl:apply-templates select="publisher"/>
      <xsl:if test="count(publisher)=1">
        <xsl:text> </xsl:text>
      </xsl:if>

      <xsl:text>. </xsl:text>

    </xsl:when>

    <!-- wenn Zeitschriftenartikel -->
    <xsl:when test="@type='article'">
      <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn inproceedings -->
    <xsl:when test="@type='inproceedings'">
      <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn in Buch -->
    <xsl:when test="@type='inbook'">
      <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn incollection -->
    <xsl:when test="@type='incollection'">
      <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn unpublished -->
    <xsl:when test="@type='unpublished'">
      <xsl:apply-templates select="@type"/>
    </xsl:when>
```



```

        <!-- wenn Techreport -->
        <xsl:when test="@type='techreport'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <xsl:when test="@type='mastersthesis'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>
</div>
</td>
</tr>

</xsl:for-each>
</xsl:if>

</xsl:if>
</table>
</xsl:if>
</xsl:if>
</xsl:if>
</xsl:if>
</xsl:if>
</xsl:if>

<!-- falls Suchbegriffe eingetragen -->

<!-- Sortieren nach Verfassernachname anschliessend nach Jahr -->

    <xsl:if test="$sort = 'lastn'">

        <xsl:if test="$order = 'descending'">

            <xsl:apply-templates select="bibitem[contains(string(.), $lastname)]">
            <xsl:sort order="descending" select="./author/lastname"/>
            <xsl:sort order="descending" select="./author/firstname"/>
            <xsl:sort order="descending" select="./year"/>
            </xsl:apply-templates>

            </xsl:if>

            <xsl:if test="$order = 'ascending'">

                <xsl:apply-templates select="bibitem[contains(string(.), $lastname)]">
                <xsl:sort order="ascending" select="./author/lastname"/>
                <xsl:sort order="ascending" select="./author/firstname"/>
                <xsl:sort order="ascending" select="./year"/>
                </xsl:apply-templates>

                </xsl:if>
            </xsl:if>
        </xsl:if>
    </xsl:if>

```

```

        <!-- Sortierung nach Jahr und anschliessend nach Autor absteigend -->
        <xsl:if test="$sort = 'year'">
            <xsl:if test="$order = 'descending'">

                <xsl:apply-templates select="bibitem[contains(string(.), $lastname)]">
                    <xsl:sort order="descending" select="./year"/>
                    <xsl:sort order="descending" select="./author/lastname"/>
                    <xsl:sort order="descending" select="./author/firstname"/>
                </xsl:apply-templates>

            </xsl:if>

            <!-- Sortierung nach Jahr und anschliessend nach Autor aufsteigend-->
            <xsl:if test="$order = 'ascending'">
                <xsl:apply-templates select="bibitem[contains(string(.), $lastname)]">
                    <xsl:sort order="ascending" select="./year"/>
                    <xsl:sort order="ascending" select="./author/lastname"/>
                    <xsl:sort order="ascending" select="./author/firstname"/>
                </xsl:apply-templates>

            </xsl:if>

        </xsl:if>

    </xsl:template>

    <xsl:template match="author">
        <xsl:choose>
            <!-- bei mehreren Autoren Komma und and einfuegen -->
            <xsl:when test="count(ancestor::bibitem/author)>1">

                <font style="font-variant:normal">
                    <xsl:value-of select="child::lastname"/>
                    <!-- wenn Vorname existiert , Vorname blank -->
                    <xsl:if test="count(firstname)=1">
                        <xsl:text>, </xsl:text>
                        <xsl:value-of select="child::firstname"/>
                        <xsl:text> </xsl:text>
                    </xsl:if>
                    <!-- wenn zweiter Vorname existiert , zweiterVorname blank -->
                    <xsl:if test="count(middlename)=1">
                        <xsl:value-of select="child::middlename"/>
                    </xsl:if>

                    <xsl:if test="position() != last()">
                        <xsl:text> and </xsl:text>
                    </xsl:if>

                </font>

            </xsl:when>

```

```
<xsl:when test="count(ancestor::bibitem/author)=1">
  <!-- bei einem Autor -->
    <font style="font-variant:normal">
      <xsl:value-of select="child::lastname"/>

      <xsl:if test="count(firstname)=1">
        <xsl:text>, </xsl:text>
        <xsl:value-of select="child::firstname"/>
        <xsl:text> </xsl:text>
      </xsl:if>

      <xsl:if test="count(middlename)=1">
        <xsl:value-of select="child::middlename"/>
        <xsl:text> </xsl:text>
      </xsl:if>

    </font>

  </xsl:when>
</xsl:choose>

</xsl:template>

<xsl:template match="title">

  <i> <xsl:value-of select="ancestor::bibitem/title"/> </i>

</xsl:template>

<xsl:template match="address">

    <font style="font-variant:normal">
      <xsl:value-of select="ancestor::bibitem/address"/>
    </font>

  </xsl:template>

<xsl:template match="publisher">
    <font style="font-variant:normal">
      <xsl:value-of select="ancestor::bibitem/publisher"/>
    </font>
  </xsl:template>

<xsl:template match="year">

    <font style="font-variant:normal">
      <xsl:value-of select="ancestor::bibitem/year"/>
    </font>

  </xsl:template>
```

```

<xsl:template match="@type">

    <xsl:choose>

        <!--      article  -->
        <xsl:when test="ancestor::bibitem[@type='article']">

            <xsl:if test="count(ancestor::bibitem/year)=1">
                <xsl:text> </xsl:text>
                <xsl:value-of select="ancestor::bibitem/year"/>
                <xsl:text>, </xsl:text>
            </xsl:if>

            <font style="normal">
            <xsl:if test="count(ancestor::bibitem/title)=1">
            <xsl:text> ' </xsl:text>
                <xsl:value-of select="ancestor::bibitem/title"/>
                <xsl:text>', </xsl:text>
            </xsl:if>
            </font>

            <xsl:if test="count(ancestor::bibitem/journal)=1">
            <xsl:text></xsl:text>
            <i><xsl:value-of select="ancestor::bibitem/journal"/> </i>
            <xsl:text></xsl:text>
            </xsl:if>

            <font style="normal">
            <xsl:if test="count(ancestor::bibitem/volume)=1">
            <xsl:text>, vol. </xsl:text>
            <xsl:value-of select="ancestor::bibitem/volume"/>
            <xsl:text></xsl:text>
            </xsl:if>

            <xsl:if test="count(ancestor::bibitem/number)=1">
            <xsl:text>, no. </xsl:text>
            <xsl:value-of select="ancestor::bibitem/number"/>
            <xsl:text></xsl:text>
            </xsl:if>

            <xsl:if test="count(ancestor::bibitem/month)=1">
            <xsl:text> - </xsl:text>
            <xsl:value-of select="ancestor::bibitem/month"/>
            </xsl:if>

            <xsl:if test="count(ancestor::bibitem/pages)=1">
            <xsl:text>, pp. </xsl:text>
            <xsl:value-of select="ancestor::bibitem/pages"/>
            <xsl:text></xsl:text>
            </xsl:if>

```

```

        </font>
    </xsl:when>

    <!--      inproceedings  -->
    <xsl:when test="ancestor::bibitem[@type='inproceedings']">

        <xsl:if test="count(ancestor::bibitem/year)=1">
            <xsl:text> </xsl:text>
            <xsl:value-of select="ancestor::bibitem/year"/>
            <xsl:text>,</xsl:text>
        </xsl:if>

        <font style="normal">
            <xsl:if test="count(ancestor::bibitem/title)=1">
                <xsl:text> '</xsl:text>
                <xsl:value-of select="ancestor::bibitem/title"/>
                <xsl:text>'</xsl:text>
            </xsl:if>
        </font>

        <xsl:if test="count(ancestor::bibitem/booktitle)=1">
            <xsl:text>,</xsl:text>
            <i><xsl:value-of select="ancestor::bibitem/booktitle"/> </i>
            <xsl:text></xsl:text>
        </xsl:if>

        <xsl:if test="count(ancestor::bibitem/series)=1">
            <xsl:text> (</xsl:text>
            <xsl:value-of select="ancestor::bibitem/series"/>
            <xsl:text>)</xsl:text>
        </xsl:if>

        <font style="normal">
            <xsl:if test="count(ancestor::bibitem/volume)=1">
                <xsl:text>,</xsl:text> vol.</xsl:text>
                <xsl:value-of select="ancestor::bibitem/volume"/>
                <xsl:text></xsl:text>
            </xsl:if>

            <xsl:if test="count(ancestor::bibitem/number)=1">
                <xsl:text>,</xsl:text> no.</xsl:text>
                <xsl:value-of select="ancestor::bibitem/number"/>
                <xsl:text></xsl:text>
            </xsl:if>

            <xsl:if test="count(ancestor::bibitem/month)=1">
                <xsl:text> - </xsl:text>
                <xsl:value-of select="ancestor::bibitem/month"/>
            </xsl:if>

            <xsl:if test="count(ancestor::bibitem/address)=1">
                <xsl:text>,</xsl:text>
                <xsl:value-of select="ancestor::bibitem/address"/>

```

```

</xsl:if>

<xsl:if test="count(ancestor::bibitem/publisher)=1">
<xsl:text>, </xsl:text>
<xsl:value-of select="ancestor::bibitem/publisher"/>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/pages)=1">
<xsl:text>, pp. </xsl:text>
<xsl:value-of select="ancestor::bibitem/pages"/>
<xsl:text></xsl:text>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/note)=1">
<xsl:text> </xsl:text>
<xsl:value-of select="ancestor::bibitem/note"/>
<xsl:text></xsl:text>
</xsl:if>
<xsl:text>.</xsl:text>
</font>
</xsl:when>

<!-- incollection -->

<xsl:when test="ancestor::bibitem[@type='incollection']">

<xsl:if test="count(ancestor::bibitem/year)=1">
<xsl:text> </xsl:text>
<xsl:value-of select="ancestor::bibitem/year"/>
<xsl:text>, </xsl:text>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/title)=1">
<font style="normal">
<xsl:text> '</xsl:text>
<xsl:value-of select="ancestor::bibitem/title"/>
<xsl:text>' </xsl:text>
</font>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/booktitle)=1">
<xsl:text>in </xsl:text>
<i><xsl:value-of select="ancestor::bibitem/booktitle"/> </i>

<xsl:if test="count(ancestor::bibitem/series)=1">
<xsl:text> (</xsl:text>
<xsl:value-of select="ancestor::bibitem/series"/>
<xsl:text>) </xsl:text>
</xsl:if>

<xsl:text>, </xsl:text>
</xsl:if>

```

```

<xsl:if test="count(ancestor::bibitem/editor)=1">
  <xsl:if test="count(ancestor::bibitem/booktitle)=0">
    <xsl:text>in </xsl:text>
  </xsl:if>
  <font style="font-variant:normal">
<xsl:value-of select="ancestor::bibitem/editor"/>
</font>
    <xsl:text>, </xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/address)=1">
    <xsl:text> </xsl:text>
    <xsl:value-of select="ancestor::bibitem/address"/>
    <xsl:text> </xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/publisher)=1">

    <xsl:value-of select="ancestor::bibitem/publisher"/>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/pages)=1">
    <xsl:text>, pp. </xsl:text>
    <xsl:value-of select="ancestor::bibitem/pages"/>
    <xsl:text></xsl:text>
  </xsl:if>

  <xsl:text>.</xsl:text>

</xsl:when>

<!--      inbook -->

  <xsl:when test="ancestor::bibitem[@type='inbook']">

    <xsl:if test="count(ancestor::bibitem/year)=1">
      <xsl:text> </xsl:text>
      <xsl:value-of select="ancestor::bibitem/year"/>
      <xsl:text>, </xsl:text>
    </xsl:if>

    <font style="normal">
      <xsl:if test="count(ancestor::bibitem/chapter)=1">
        <xsl:text> '</xsl:text>
        <xsl:value-of select="ancestor::bibitem/chapter"/>

        <xsl:text>' </xsl:text>
      </xsl:if>
    </font>

    <xsl:if test="count(ancestor::bibitem/editor)=1">
      <xsl:text>in ed.</xsl:text>
      <font style="font-variant:normal">

```

```

<xsl:value-of select="ancestor::bibitem/editor"/>
</font>
<xsl:text>. </xsl:text>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/title)=1">
<xsl:if test="count(ancestor::bibitem/editor)=0">
<xsl:text>in </xsl:text>
</xsl:if>
<i><xsl:value-of select="ancestor::bibitem/title"/> </i>

<xsl:if test="count(ancestor::bibitem/series)=1">
<xsl:text> (</xsl:text>
<xsl:value-of select="ancestor::bibitem/series"/>
<xsl:text>) </xsl:text>
</xsl:if>

<xsl:text>, </xsl:text>
</xsl:if>

<font style="normal">
<xsl:if test="count(ancestor::bibitem/volume)=1">
<xsl:text>vol. </xsl:text>
<xsl:value-of select="ancestor::bibitem/volume"/>

<xsl:text></xsl:text>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/number)=1">
<xsl:text>, no. </xsl:text>
<xsl:value-of select="ancestor::bibitem/number"/>
<xsl:text></xsl:text>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/month)=1">
<xsl:text> - </xsl:text>
<xsl:value-of select="ancestor::bibitem/month"/>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/publisher)=1">
<xsl:text>, </xsl:text>
<xsl:value-of select="ancestor::bibitem/publisher"/>
<xsl:text>, </xsl:text>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/address)=1">
<xsl:text></xsl:text>
<xsl:value-of select="ancestor::bibitem/address"/>
<xsl:text>, </xsl:text>
</xsl:if>

<xsl:if test="count(ancestor::bibitem/pages)=1">
<xsl:text>, pp. </xsl:text>

```



```

        <xsl:value-of select="ancestor::bibitem/pages"/>
        <xsl:text></xsl:text>
      </xsl:if>

    </font>
  </xsl:when>

  <!--      mastersthesis  -->

<xsl:when test="ancestor::bibitem[@type='mastersthesis']">

  <xsl:if test="count(ancestor::bibitem/year)=1">
    <xsl:text>, </xsl:text>
    <xsl:value-of select="ancestor::bibitem/year"/>
    <xsl:text>, </xsl:text>
  </xsl:if>

  <font style="font-variant:normal">
    <xsl:if test="count(ancestor::bibitem/title)=1">
      <xsl:text> '</xsl:text>
      <xsl:value-of select="ancestor::bibitem/title"/>

      <xsl:text>' </xsl:text>
    </xsl:if>
  </font>

  <xsl:if test="count(ancestor::bibitem/type)=1">
    <xsl:text>, </xsl:text>
    <xsl:value-of select="ancestor::bibitem/type"/>
    <xsl:text></xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/school)=1">
    <xsl:text>, </xsl:text>
    <xsl:value-of select="ancestor::bibitem/school"/>
    <xsl:text>, </xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/address)=1">
    <xsl:text></xsl:text>
    <xsl:value-of select="ancestor::bibitem/address"/>
    <xsl:text></xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/pages)=1">
    <xsl:text>, pp. </xsl:text>
    <xsl:value-of select="ancestor::bibitem/pages"/>
    <xsl:text></xsl:text>
  </xsl:if>
  <xsl:text>.</xsl:text>

</xsl:when>

  <!--      unpublished  -->

```

```

<xsl:when test="ancestor::bibitem[@type='unpublished']">
  <font style="font-variant:normal">
    <xsl:value-of select="ancestor::bibitem/title"/>
    <xsl:if test="count(ancestor::bibitem/title)=1">
      <xsl:text>. </xsl:text>
    </xsl:if>
  </font>

  <xsl:if test="count(ancestor::bibitem/note)=1">
    <xsl:text>Note: </xsl:text>
    <font style="font-variant:normal">
      <xsl:value-of select="ancestor::bibitem/note"/>
    </font>
    <xsl:text> - Unver&#246;ffentlicht </xsl:text>
  </xsl:if>

</xsl:when>

<!--      techreport -->
<xsl:when test="ancestor::bibitem[@type='techreport']">

  <xsl:if test="count(ancestor::bibitem/year)=1">
    <xsl:text> </xsl:text>
    <xsl:value-of select="ancestor::bibitem/year"/>
    <xsl:text>, </xsl:text>
  </xsl:if>

  <font style="normal">
    <xsl:if test="count(ancestor::bibitem/title)=1">
      <xsl:text> </xsl:text>
    <i><xsl:value-of select="ancestor::bibitem/title"/></i>
    <xsl:text>, </xsl:text>
  </xsl:if>
</font>

  <xsl:if test="count(ancestor::bibitem/editor)=1">
    <xsl:text> </xsl:text>
    <font style="font-variant:normal">
      <xsl:value-of select="ancestor::bibitem/editor"/>
    </font>
    <xsl:text>: </xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/institution)=1">
    <xsl:text> </xsl:text>
    <font style="font-variant:normal">
      <xsl:value-of select="ancestor::bibitem/institution"/>
    </font>
    <xsl:text> </xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/booktitle)=1">
    <xsl:if test="count(ancestor::bibitem/editor)=0">

```

```

        <xsl:text>in </xsl:text>
      </xsl:if>
    <i><xsl:value-of select="ancestor::bibitem/booktitle"/> </i>
    <xsl:text> </xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/series)=1">
    <xsl:text> (</xsl:text>
    <xsl:value-of select="ancestor::bibitem/series"/>
    <xsl:text>) </xsl:text>
  </xsl:if>

  <font style="normal">
    <xsl:if test="count(ancestor::bibitem/volume)=1">
      <xsl:text>.</xsl:text>
      <xsl:value-of select="ancestor::bibitem/volume"/>

    <xsl:text> </xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/number)=1">
    <xsl:text>,</xsl:text>
    <xsl:value-of select="ancestor::bibitem/number"/>
    <xsl:text></xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/month)=1">
    <xsl:text> - </xsl:text>
    <xsl:value-of select="ancestor::bibitem/month"/>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/address)=1">
    <xsl:text> </xsl:text>
    <xsl:value-of select="ancestor::bibitem/address"/>
    <xsl:text> </xsl:text>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/publisher)=1">
    <xsl:value-of select="ancestor::bibitem/publisher"/>
  </xsl:if>

  <xsl:if test="count(ancestor::bibitem/pages)=1">
    <xsl:text>,</xsl:text> pp. </xsl:text>
    <xsl:value-of select="ancestor::bibitem/pages"/>
    <xsl:text></xsl:text>
  </xsl:if>

  <xsl:text> - Technical Report</xsl:text>
</font>
  </xsl:when>

</xsl:choose>

```

```
</xsl:template>

<xsl:template match="bibitem">

  <!-- HIER IST DIE SUCHE -->

<xsl:choose>

  <!-- keine Einschraenkung bei Kind -->
    <xsl:when test="string-length($kind)=0">

      <!-- Titel und Jahr vorhanden -->
      <xsl:if test="string-length($title)>0">
        <xsl:if test="title[contains(string(.), $title)]">
          <xsl:if test="string-length($year)>0">
            <xsl:if test="year[contains(string(.), $year)]">
              <xsl:if test="author[contains(string(./lastname), $lastname)]">
                <table>
                  <tr>
                    <td>
                      <div>

                        <xsl:apply-templates select="author"/>

                        <xsl:choose>

                          <!-- wenn selbstaendige Einheit -->
                          <xsl:when test="@type='book'">
                            <xsl:apply-templates select="year"/>
                            <xsl:text>, </xsl:text>

                            <xsl:if test="count(title)=1">
                              <xsl:apply-templates select="title"/>
                              <xsl:text>.</xsl:text>
                            </xsl:if>

                            <xsl:if test="count(address)=1">
                              <xsl:apply-templates select="address"/>
                              <xsl:text>:</xsl:text>
                            </xsl:if>

                            <xsl:if test="count(publisher)=1">
                              <xsl:apply-templates select="publisher"/>
                              </xsl:if>
                            <xsl:text>.</xsl:text>

                        </xsl:when>

                          <!-- wenn Zeitschriftenartikel -->
                          <xsl:when test="@type='article'">
                            <xsl:apply-templates select="@type"/>
                            </xsl:when>

                        </div>
                    </td>
                  </tr>
                </table>
              </xsl:if>
            </xsl:if>
          </xsl:if>
        </xsl:if>
      </xsl:if>
    </xsl:when>
  </xsl:choose>
</xsl:template>
```

```

        <!-- wenn inproceedings -->
        <xsl:when test="@type='inproceedings'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn in Buch -->
        <xsl:when test="@type='inbook'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn incollection -->
        <xsl:when test="@type='incollection'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn unpublished -->
        <xsl:when test="@type='unpublished'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn Techreport -->
        <xsl:when test="@type='techreport'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <xsl:when test="@type='mastersthesis'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>

</div>
</td>
</tr>
</table>
</xsl:if>

</xsl:if>
</xsl:if>
<!-- Titel vorhanden aber kein Jahr -->
    <xsl:if test="string-length($year)=0">
    <xsl:if test="author[contains(string(/.lastname), $lastname)]">

        <table>
        <tr><td><div>

            <xsl:apply-templates select="author"/>

            <xsl:choose>

            <!-- wenn selbstaendige Einheit -->
            <xsl:when test="@type='book'">
            <xsl:apply-templates select="year"/>

```

```

<xsl:text>, </xsl:text>

<xsl:if test="count(title)=1">
<xsl:apply-templates select="title"/>
<xsl:text>. </xsl:text>
</xsl:if>

<xsl:if test="count(address)=1">
<xsl:apply-templates select="address"/>
<xsl:text>: </xsl:text>
</xsl:if>

<xsl:if test="count(publisher)=1">
<xsl:apply-templates select="publisher"/>
</xsl:if>
<xsl:text>. </xsl:text>

</xsl:when>

<!-- wenn Zeitschriftenartikel -->
<xsl:when test="@type='article'">
  <xsl:apply-templates select="@type"/>
</xsl:when>

<!-- wenn inproceedings -->
<xsl:when test="@type='inproceedings'">
  <xsl:apply-templates select="@type"/>
</xsl:when>

<!-- wenn in Buch -->
<xsl:when test="@type='inbook'">
  <xsl:apply-templates select="@type"/>
</xsl:when>

<!-- wenn incollection -->
<xsl:when test="@type='incollection'">
  <xsl:apply-templates select="@type"/>
</xsl:when>

<!-- wenn unpublished -->
<xsl:when test="@type='unpublished'">
  <xsl:apply-templates select="@type"/>
</xsl:when>

<!-- wenn Techreport -->
<xsl:when test="@type='techreport'">
  <xsl:apply-templates select="@type"/>
</xsl:when>

<xsl:when test="@type='mastersthesis'">
  <xsl:apply-templates select="@type"/>
</xsl:when>
</xsl:choose>

```

```

        </div> </td></tr>
    </table>

</xsl:if>
</xsl:if>
</xsl:if>
</xsl:if>
    <!-- Jahr vorhanden aber kein Titel -->
<xsl:if test="string-length($title)=0">
    <xsl:if test="string-length($year )>0">
        <xsl:if test="year[contains(string(.), $year)]">
            <xsl:if test="author[contains(string(./lastname), $lastname)]">

                <table>

<tr><td><div>

    <xsl:apply-templates select="author"/>

    <xsl:choose>

        <!-- wenn selbstaendige Einheit -->
        <xsl:when test="@type='book'">
            <xsl:apply-templates select="year"/>
            <xsl:text>, </xsl:text>

            <xsl:if test="count(title)=1">
                <xsl:apply-templates select="title"/>
                <xsl:text>.. </xsl:text>
            </xsl:if>

            <xsl:if test="count(address)=1">
                <xsl:apply-templates select="address"/>
                <xsl:text>: </xsl:text>
            </xsl:if>

            <xsl:if test="count(publisher)=1">
                <xsl:apply-templates select="publisher"/>
            </xsl:if>
            <xsl:text>.. </xsl:text>

        </xsl:when>

        <!-- wenn Zeitschriftenartikel -->
        <xsl:when test="@type='article'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn inproceedings -->
        <xsl:when test="@type='inproceedings'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn in Buch -->
        <xsl:when test="@type='inbook'">
            <xsl:apply-templates select="@type"/>

```

```

        </xsl:when>

        <!-- wenn incollection -->
        <xsl:when test="@type='incollection'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn unpublished -->
        <xsl:when test="@type='unpublished'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn Techreport -->
        <xsl:when test="@type='techreport'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <xsl:when test="@type='mastersthesis'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>

</div> </td></tr></table>
</xsl:if>
</xsl:if>
</xsl:if>
<!-- kein Titel und kein Jahr Testvariable ist leer -->
<xsl:if test="string-length($test )=0">
<xsl:if test="string-length($year )=0">
    <xsl:if test="author[contains(string(/.lastname), $lastname)]">

        <table>
<tr><td><div>

        <xsl:apply-templates select="author"/>

        <xsl:choose>

        <!-- wenn selbstaendige Einheit -->
        <xsl:when test="@type='book'">
        <xsl:apply-templates select="year"/>
        <xsl:text>, </xsl:text>

        <xsl:if test="count(title)=1">
        <xsl:apply-templates select="title"/>
        <xsl:text>. </xsl:text>
        </xsl:if>

        <xsl:if test="count(address)=1">
        <xsl:apply-templates select="address"/>
        <xsl:text>: </xsl:text>
        </xsl:if>
    
```



```

        <xsl:if test="count(publisher)=1">
        <xsl:apply-templates select="publisher"/>
        </xsl:if>
        <xsl:text>. </xsl:text>

    </xsl:when>

    <!-- wenn Zeitschriftenartikel -->
    <xsl:when test="@type='article'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn inproceedings -->
    <xsl:when test="@type='inproceedings'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn in Buch -->
    <xsl:when test="@type='inbook'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn incollection -->
    <xsl:when test="@type='incollection'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn unpublished -->
    <xsl:when test="@type='unpublished'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn Techreport -->
    <xsl:when test="@type='techreport'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <xsl:when test="@type='mastersthesis'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    </xsl:choose>

    </div> </td></tr></table>
</xsl:if>
</xsl:if>

</xsl:if>
</xsl:if>

</xsl:when>

<!-- Hier Kind-Suche -->

```

```

<xsl:when test="string-length($kind )>0">

  <!-- Typeueberpruefung Kind -->
  <xsl:if test="@type= $kind">

    <!-- titel und jahr vorhanden -->
    <!-- <xsl:if test="string-length($lastname)=0"> -->

    <xsl:if test="string-length($title)>0">
      <xsl:if test="title[contains(string(.), $title)]">
        <xsl:if test="string-length($year )>0">
          <xsl:if test="year[contains(string(.), $year)]">
            <xsl:if test="author[contains(string(/lastname), $lastname)]">
              <table>
                <tr><td><div>

                  <xsl:apply-templates select="author"/>

                  <xsl:choose>

                    <!-- wenn selbstaendige Einheit -->
                    <xsl:when test="@type='book'">
                      <xsl:apply-templates select="year"/>
                      <xsl:text>, </xsl:text>

                      <xsl:if test="count(title)=1">
                        <xsl:apply-templates select="title"/>
                        <xsl:text>. </xsl:text>
                      </xsl:if>

                      <xsl:if test="count(address)=1">
                        <xsl:apply-templates select="address"/>
                        <xsl:text>: </xsl:text>
                      </xsl:if>

                      <xsl:if test="count(publisher)=1">
                        <xsl:apply-templates select="publisher"/>
                      </xsl:if>
                      <xsl:text>. </xsl:text>

                    </xsl:when>

                    <!-- wenn Zeitschriftenartikel -->
                    <xsl:when test="@type='article'">
                      <xsl:apply-templates select="@type"/>
                    </xsl:when>

                    <!-- wenn inproceedings -->
                    <xsl:when test="@type='inproceedings'">
                      <xsl:apply-templates select="@type"/>
                    </xsl:when>

                    <!-- wenn in Buch -->

```



```

        </xsl:if>

        <xsl:if test="count(publisher)=1">
        <xsl:apply-templates select="publisher"/>
        </xsl:if>
        <xsl:text>. </xsl:text>

    </xsl:when>

    <!-- wenn Zeitschriftenartikel -->
    <xsl:when test="@type='article'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn inproceedings -->
    <xsl:when test="@type='inproceedings'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn in Buch -->
    <xsl:when test="@type='inbook'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn incollection -->
    <xsl:when test="@type='incollection'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn unpublished -->
    <xsl:when test="@type='unpublished'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <!-- wenn Techreport -->
    <xsl:when test="@type='techreport'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    <xsl:when test="@type='mastersthesis'">
        <xsl:apply-templates select="@type"/>
    </xsl:when>

    </xsl:choose>
</div> </td></tr>

</table>
<!-- </xsl:if> -->
</xsl:if>
</xsl:if>
</xsl:if>

</xsl:if>
</xsl:if>

```

```

        <!-- kein titel aber jahr vorhanden -->
<xsl:if test="string-length($title)=0">
    <xsl:if test="string-length($year )>0">
        <xsl:if test="year[contains(string(.), $year)]">
<xsl:if test="author[contains(string(/lastname), $lastname)]">
            <table>
                <tr><td><div>

                    <xsl:apply-templates select="author"/>

                    <xsl:choose>

<!-- wenn selbstaendige Einheit -->
                        <xsl:when test="@type='book'">
                            <xsl:apply-templates select="year"/>
                            <xsl:text>, </xsl:text>

                            <xsl:if test="count(title)=1">
                                <xsl:apply-templates select="title"/>
                                <xsl:text> . </xsl:text>
                            </xsl:if>

                            <xsl:if test="count(address)=1">
                                <xsl:apply-templates select="address"/>
                                <xsl:text>: </xsl:text>
                            </xsl:if>

                            <xsl:if test="count(publisher)=1">
                                <xsl:apply-templates select="publisher"/>
                                </xsl:if>
                                <xsl:text> . </xsl:text>

                        </xsl:when>

                        <!-- wenn Zeitschriftenartikel -->
                        <xsl:when test="@type='article'">
                            <xsl:apply-templates select="@type"/>
                        </xsl:when>

                        <!-- wenn inproceedings -->
                        <xsl:when test="@type='inproceedings'">
                            <xsl:apply-templates select="@type"/>
                        </xsl:when>

                        <!-- wenn in Buch -->
                        <xsl:when test="@type='inbook'">
                            <xsl:apply-templates select="@type"/>
                        </xsl:when>

                        <!-- wenn incollection -->
                        <xsl:when test="@type='incollection'">
                            <xsl:apply-templates select="@type"/>
                        </xsl:when>

```

```

        <!-- wenn unpublished -->
        <xsl:when test="@type='unpublished'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn Techreport -->
        <xsl:when test="@type='techreport'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <xsl:when test="@type='mastersthesis'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>

</div> </td></tr></table>

</xsl:if>
</xsl:if>
</xsl:if>
    <!-- kind vorhanden aber kein titel kein jahr -->
    <xsl:if test="string-length($year )=0">
        <xsl:if test="@type= $kind">
            <xsl:if test="author [contains(string(/. /lastname), $lastname)]">
                <table>
                    <tr><td><div>

<xsl:apply-templates select="author"/>

<xsl:choose>

<!-- wenn selbstaendige Einheit -->
        <xsl:when test="@type='book'">
        <xsl:apply-templates select="year"/>
        <xsl:text>, </xsl:text>

        <xsl:if test="count(title)=1">
        <xsl:apply-templates select="title"/>
        <xsl:text>.. </xsl:text>
        </xsl:if>

        <xsl:if test="count(address)=1">
        <xsl:apply-templates select="address"/>
        <xsl:text>: </xsl:text>
        </xsl:if>

        <xsl:if test="count(publisher)=1">
        <xsl:apply-templates select="publisher"/>
        </xsl:if>
        <xsl:text>.. </xsl:text>

    </xsl:when>

```

```

        <!-- wenn Zeitschriftenartikel -->
        <xsl:when test="@type='article'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn inproceedings -->
        <xsl:when test="@type='inproceedings'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn in Buch -->
        <xsl:when test="@type='inbook'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn incollection -->
        <xsl:when test="@type='incollection'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn unpublished -->
        <xsl:when test="@type='unpublished'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn Techreport -->
        <xsl:when test="@type='techreport'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

        <xsl:when test="@type='mastersthesis'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>

    </div> </td></tr></table>
</xsl:if>
</xsl:if>
</xsl:if>
<!-- </xsl:if> -->
<!-- kind vorhanden titel vorhanden jahr vorhanden-->

<!-- <xsl:if test="string-length($lastname)>0">
    <xsl:if test="author/lastname[contains(string(.), $lastname)]"> -->
    <xsl:if test="string-length($title)>0">
        <xsl:if test="title[contains(string(.), $title)]">
            <xsl:if test="string-length($year )>0">
                <xsl:if test="year[contains(string(.), $year)]">
                    <xsl:if test="author [contains(string(/lastname), $lastname)]">
                        <table>
                        <tr><td><div>

                            <xsl:apply-templates select="author"/>

```

```
<xsl:choose>

<!-- wenn selbstaendige Einheit -->
  <xsl:when test="@type='book'">
    <xsl:apply-templates select="year"/>
    <xsl:text>, </xsl:text>

    <xsl:if test="count(title)=1">
      <xsl:apply-templates select="title"/>
      <xsl:text>. </xsl:text>
    </xsl:if>

    <xsl:if test="count(address)=1">
      <xsl:apply-templates select="address"/>
      <xsl:text>: </xsl:text>
    </xsl:if>

    <xsl:if test="count(publisher)=1">
      <xsl:apply-templates select="publisher"/>
    </xsl:if>
    <xsl:text>. </xsl:text>

  </xsl:when>

  <!-- wenn Zeitschriftenartikel -->
  <xsl:when test="@type='article'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn inproceedings -->
  <xsl:when test="@type='inproceedings'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn in Buch -->
  <xsl:when test="@type='inbook'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn incollection -->
  <xsl:when test="@type='incollection'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn unpublished -->
  <xsl:when test="@type='unpublished'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>

  <!-- wenn Techreport -->
  <xsl:when test="@type='techreport'">
    <xsl:apply-templates select="@type"/>
  </xsl:when>
```



```

        <xsl:when test="@type='mastersthesis'">
            <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>

    </div> </td></tr></table>
</xsl:if>
</xsl:if>
</xsl:if>

<!-- kind vorhanden  titel vorhanden kein jahr-->
<!-- Titel vorhanden kein Jahr -->
    <xsl:if test="string-length($year )=0">
        <xsl:if test="author [contains(string(/. /lastname), $lastname)]">

            <table>
            <tr><td><div>

                <xsl:apply-templates select="author"/>

                <xsl:choose>

                    <!-- wenn selbstaendige Einheit -->
                    <xsl:when test="@type='book'">
                        <xsl:apply-templates select="year"/>
                        <xsl:text>,< /xsl:text>

                        <xsl:if test="count(title)=1">
                            <xsl:apply-templates select="title"/>
                            <xsl:text>.< /xsl:text>
                        </xsl:if>

                        <xsl:if test="count(address)=1">
                            <xsl:apply-templates select="address"/>
                            <xsl:text>:< /xsl:text>
                        </xsl:if>

                        <xsl:if test="count(publisher)=1">
                            <xsl:apply-templates select="publisher"/>
                        </xsl:if>
                        <xsl:text>.< /xsl:text>

                    </xsl:when>

                    <!-- wenn Zeitschriftenartikel -->
                    <xsl:when test="@type='article'">
                        <xsl:apply-templates select="@type"/>
                    </xsl:when>

                    <!-- wenn inproceedings -->
                    <xsl:when test="@type='inproceedings'">
                        <xsl:apply-templates select="@type"/>

```

```

        </xsl:when>

        <!-- wenn in Buch -->
        <xsl:when test="@type='inbook'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn incollection -->
        <xsl:when test="@type='incollection'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn unpublished -->
        <xsl:when test="@type='unpublished'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <!-- wenn Techreport -->
        <xsl:when test="@type='techreport'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

        <xsl:when test="@type='mastersthesis'">
        <xsl:apply-templates select="@type"/>
        </xsl:when>

    </xsl:choose>
</div> </td></tr></table>

    </xsl:if>
  </xsl:if>
</xsl:if>
</xsl:if>
</xsl:if>
</xsl:if>

</xsl:when>

</xsl:choose>

</xsl:template>

</xsl:stylesheet>

```

Anhang C

Kompletter Code des PHP-Formulars

```
<html>

<head>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
<LINK REL=STYLESHEET TYPE="text/css" href="style.css"?>
<style type="text/css">

#rahmen {
    border-width:5px;
    border-style:solid;
    border-color:#007f97;
    padding:1cm;
    margin-left:1cm;
    text-align:justify; }
}

</style>

<title>Literaturrecherche
</title>

</head>

<body>
<a name="up"> <table border="0"></a>

<tr>
<td width="30px">
</td>
<td >

<table>
<tr>
<td id="rahmen">
<font face="Arial">

<h1>Literaturrecherche</h1>

<?php

if ($bib == "xml") {$bibname = "bibliography.xml";}
if ($style == "DIN") {$xslname = "din.xsl";}
if ($style == "APA") {$xslname = "apa.xsl";}
if ($style == "Harvard") {$xslname = "harvard.xsl";}
```

```

if ($sortierung == "Autornamen") {$sort = "lastn";}
if ($sortierung == "Jahr") {$sort = "year";}
if ($reihe == "aufsteigend") {$order = "ascending";}
if ($reihe == "absteigend") {$order = "descending";}

if ($lastname or $title or $year or $kind) {$fields = "";} else {$fields
="1";}

if ($sent and $style) {
echo "<h3>Sie haben folgende Auswahl getroffen:</h3><p>";
echo "<b>Autornamen:</b> $lastname<br>";
echo "<b>Titel:</b> $title<br>";
echo "<b>Erscheinungsjahr:</b> $year<br>";
echo "<b>Literaturart:</b> $kind<br>";
//echo "<b>verwendete Bibliographie:</b> $bib<br>";
echo "<b>Sortierung:</b> $sortierung<br>";
echo "<b>Reihenfolge:</b> $reihe<br>";
echo "<b>Anzeigestil:</b> $style<br></p>";
echo " <a href=\"javascript:history.back()\"><img src=\"back.gif\"
border=\"0\"></a> neue Suche</p>";

$XH = xslt_create();

$param = array ('kind' => $kind, 'lastname' => $lastname, 'title' => $title,
'year' => $year, 'test' => $fields, 'sort' =>$sort, 'order' =>$order );

$result = xslt_process($XH, $bibname, $xslname, NULL, array(), $param);

if ($result) {
    print "<hr noshade>";
    print "<div>\n";
    print $result;
    print "</div>\n";
    print "<hr noshade>";
}
else {
    print "Sorry, sample.xml could not be transformed by sample.xsl into";
    print " the \"$result\" variable the reason is that " . xslt_error($XH) . "
and the ";
    print "error code is " . xslt_errno($XH);
}

xslt_free($XH);

print " <p> <a href =\"#up\"><img src=\"up.gif\" border=\"0\"></a> Anfang \n
";
print " <a href=\"javascript:history.back()\"><img src=\"back.gif\"
border=\"0\"></a> neue Suche</p>";

}

```

```

else {

echo <<<FORM

<h3>Literatursuche und Bibliographieausgabe</h3>
<form method="get" action="$PHP_SELF" >

<font face="Arial">
<p>
<p>
Literatursuche <font size="-1">&#42;Gro&szlig;- und Kleinschreibung
beachten&#41;</font><br>
<table border=0 width="50%">

    <tr><td>Autornamen&#42;</td><td align="left"><input type="text" size="50"
name="lastname"></td></tr>
    <tr><td width="50">Titel</td><td align="left"><input type="text"
size="50" name="title"></td></tr>
    <tr><td width="50">Jahr</td><td align="left"><input type="text" size="50"
name="year"></td></tr>

</table>
<font size="-1">&#42; nur Nachname</font>
</p>
<p>
<table width="50%">
<tr>
<td width="">
<p>Art der Quellen<br>
<select name="kind">
<option selected="selected" value="">alle</option>
<option value="book">Buch</option>
<option value="article">Zeitschriftenartikel</option>
<option value="inbook">Teil eines Buches</option>
<option value="inproceedings">Teil eines Tagungsberichtes</option>
<option value="incollection">Teil einer Aufsatzsammlung</option>
<option value="techreport">Techreport</option>
<option value="mastersthesis">Diplomarbeit</option>
<option value="unpublished">unver&#246;ffentlicht</option>
</select>

<!-- (alle Bibliographieeintr&#228;ge: ohne zus&#228;tzlichen Eintrag bei Autor,
Titel oder Jahr) -->
</p>
</td>
<td width="">
<p>Sortierung<br>
<select name="sortierung">
<option selected="selected" value="Autornamen">Autornamen</option>
<option value="Jahr">Erscheinungsjahr</option>
</select>
</p>
</td>

```

```

<td width="">
<p>Reihenfolge<br>
<select name="reihe">
<option selected="selected" value="aufsteigend">aufsteigend</option>
<option value="absteigend">absteigend</option>
</select>
</p>
</td>
</tr>
</table>
</p>
</p>
</font>

```

```

<p>Anzeigestil<br>

```

```

<input type="radio" name="style" value="DIN" checked>DIN 1505
<input type="radio" name="style" value="APA">APA
<input type="radio" name="style" value="Harvard">Harvard
</p>

```

```

<p>
<input type="hidden" name="bib" value="xml">

```

```

</p>

```

```

<p>
<input type="submit" value="Absenden">
<input type="reset" value="Zurücksetzen">
<input type="hidden" name="sent" value=1>

```

```

</p>
</font>
</font>
</form>
FORM;

```

```

}
?>

```

```

</td>
</tr>
</table>

```

```

</td>
</tr>
</table>
</body>
</html>

```

Literaturverzeichnis

Printmedien

- American Psychological Association 2001** AMERICAN PSYCHOLOGICAL ASSOCIATION: *Publication Manual of the American Psychological Association*. 5. Aufl. Washington, DC : American Psychological Association (APA), 2001 – ISBN 1-55798-790-4
- Goldfarb 2002** GOLDFARB, CHARLES F. ; PRESCOD, PAUL : *Charles F. Goldfarb's XML Handbook*. Upper Saddle River, New Jersey: Prentice Hall, 2002
- Harold 2002** HAROLD, ELLIOTTE RUSTY : *Die XML-Bibel*. 2. Aufl. Bonn: mitp-Verlag 2002 – ISBN 3-82-66-0821-6
- Lurati 2001** Lurati, Brenno ; Previtali, Luca : *BibTeXXML*. Zürich: Eidgenössische Technische Hochschule, Institut für Technische Informatik und Kommunikationsnetze, Dipl.-Arb., 2001, Zugriff auf PDF-Datei von URL <http://dret.net/netdret/docs/dalurati-previtali.pdf> (Zugriff am 3.10.2002)
- Müller 2001** Müller, Heike : *Erstellung von Bibliographien auf der Basis von XML und XSL*. Stuttgart: Fachhochschule Stuttgart – Hochschule der Medien, Fachbereich Information und Kommunikation, Dipl.-Arb., 2001

Elektronische Ressourcen

- The Bibtex-XML-HTML Project 2002** THE BIBTEX-XML-HTML PROJECT: URL <http://www.authopilot.com/xml/index.html> – Aktualisierungsdatum: 09.09.2002 (Zugriff am 3.10.2002)
- The Collection of Computer Science Bibliographies 2002** THE COLLECTION OF COMPUTER SCIENCE BIBLIOGRAPHIES: URL <http://iinwww.ira.uka.de/bibliography> (Zugriff am 3.10.2002)
- Cybrary 2001** THE UNIVERSITY OF QUEENSLAND LIBRARY: *References/Bibliography Harvard Style: Quick Guide - How to use it*. URL <http://www.library.uq.edu.au/training/citation/harvard.html> - Aktualisierungsdatum: 7.06.2001 (Zugriff am 18.09.2002)
- Kuhlmann 2002** KUHLMANN, MARCO : *BibTeXXML*. URL <http://www.ps.uni-sb.de/~kuhlmann/bibtexml> (Zugriff am 3.10.2002)
- Lorenzen 1997** LORENZEN, KLAUS F.: *Das Literaturverzeichnis in wissenschaftlichen Arbeiten: Erstellung bibliographischer Belege nach DIN 1505 Teil 2*. URL <http://www.bui.fh-hamburg.de/pers/klaus.lorenzen/ASP/litverz.pdf> (Zugriff am 15.09.2002)

- OWL 2001** OWL-ONLINE WRITING LAB AT PURDUE UNIVERSITY : *Using American Psychological Association (APA) Format (Updated to 5th Edition)*. URL http://owl.english.purdue.edu/handouts/research/r_apa.html (Zugriff am 3.10.2002)
- Patashnik 1988** PATASHNIK, OREN : *BibTeXing*. PDF-Datei. URL <http://bibtexml.sourceforge.net/btxdoc.pdf> (Zugriff am 3.10.2002)
- Real-Time Systems Laboratory 2001** REAL-TIME SYSTEMS LABORATORY: Publications. URL <http://pertsserver.cs.uiuc.edu/papers/index.html> (Zugriff am 3.10.2001)

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ort, Datum

Unterschrift